mathe II

david vajda

15. November 2026

als nächstes kommen: die algorithmen. richtiger weise, fällt mir auf beim installieren von

```
./configure
make
make install
```

da stellen sich manche leute was kompliziertes drunter vor. das sei ihr recht. wie unter

```
lex
yacc
\end{lstlisng}
oder unter
\begin{lstlisting}
lex
yacc
bison
apt-get
\end{lstlisng}
na ja, so kompliziert ist die sache nicht. das sind nutzerprogramme, die genutz
\begin{lstlisting}
make
\end{lstlisting]
ist ein programm und landet mit
\begin{lstlisting}
gcc
```

kurz den gcc-toolchain einfach im normalen default install verzeichnis. dazu gibt es einen std-path. bei linux distributionen erledigen sich die probleme über

```
apt-get
/etc/apt/sources.lst
```

kurz einer allgemeinen installations binary verwaltung. nun ist das ein problem des arbeitsprozesss. darum geht es.

- bei arbeitsprozessen
- INTERRUPT - H . . . Lenin - seit neuestem HL - ausser bei Disco Sux und hüpfenden Leuten - sie sehen wohin das führt
- sind die schritte generell nciht geheim zu halten

- we should talk in english
- so do it again? what is this? for an not german people - we have to use english - 1st
- the second is - a working process has to be - written - it must documentated
- so: if we are a laboratorium
- let me explain - if you don't the english word for laboratorium - the thing is - you need to take the german - you can speek in - indian/sanskrit what ever it is ... so you know one word - and every word without the sanscrit has to be in german

- you speak in german like a swabian and lot's of words are not really correct - not because of the swabian - but this will make them thinking and work again
- but: the problem: you jobcenter says - and comes to you world wild and the working ministers stands for youre door - what a service - and says: the problem: the problem the intension must fit
- now the problem is: if poeple sounds very like the working man, using - the punishment, making poeple working, this is the one thing - it deathly not to do anything
- but - if these persons speak in german, and he thinks all german hate him, he uses not the hole package/the hole glory of the things - he could be by 7.000.000.000 poeple
- you have to speak in english
- this doesn't means - that - you don't learn indian, africans, russian, hebrew, arabian - but this is problem
- the next thing is: these words maybe are unnecessary - but they are not in this case, cause: you: have to write down the working process - the working flow ...
- cause: a laboratory must: documentate the hole things
- it is unnessecarry to speak about what we are speaking - but the problem is: it is unnesseray we thing to speak about -

  ```
  gnu
  bison
  yacc
  lex
  gcc
  apt-get
  ```

  maybe. the thing is: here is the: distribution - a distribution a word i had to learn - is giving the poeple the product maybe. sorry it is not good for all people world wide to sleep in. and: we have to speak in russian, chinese,

arabian, indian, but we have our personal or not personal problems, and we do it without pressure, so we have to - do the things and while that, we laerning, counting

- next: the programms, binarys :

```
gnu
bison
yacc
make
```

these sounds - I am idiot. because maybe we can't use them. but this is a little strange

```
AVR
AVM


SOIC
SOC
```

sometimes you must accept these rulse. cause. but. wait, wait, wait! the problem is! i know cause, i am programming in C and i know there is a little more than a C Compiler. the is ld - the linking programm. ok. but: you have to get a little feeling. and you don't need it. from the social context, you have to make all people frightend - the will follow - it is a little like social darwinism. so you if lay - not like neo liberalism. faster, higher,... so on, the rest sleeps in. the is an economic cirses, not the money - the poeple are sleeping working technical while they are damaging the things. so we go on.
you have to have a little feeling. but: you can learn what is.

```
./configure
make
make install
```

but, you have to like apt-get. but: make is a binary - this is not a mystheric religious satanic input. this is programm like the boot loader. like everything else, packages, binaries. you have to use. ts this neccesary for the working process? yes - this is laboratory. cause.

before you start, working - with turn on the other computer, poeple can think - you are psychologist - it is ok - but you have to make thing real - the difference is between - laboratory and studies. studies will to be a laboratorium. and this is working - without dealing with money. so, the thing is, this is a apple computer, iBook. and a like linus, but i lika apple. and the other thing - there are other operating systems like

```
plan9
ms--dos
cp/m
```

what ever. there are cool. i am happy to have plan9 operating systems. but in these case they are unixes, like this! Caution microsoft producses a linux version and a UNIX. But! the thing of apt-get is that i use

```
latex2html
pdflatex
gcc
nasm
```

and so on. and there is a hard pc - a big pc. and there is this apple laptop. and if i want to make people see these things in the morning, i could stand up and go to the PC. and make it on. and i can like a little bit unprofessional working making here making there - i don't know word. but this is not the process of the laborary. but it is a process. to write it down. so i write down - that i don't make on the pc - i use the laptop. and it is a process. that i am not making deals with

```
make
gcc
as
nasm
```

i can but not on this position. you have to learn it for you brain, like it works. it is part of programming. the part of laboritry i to write it down. but what have i done. i try to install latex2html but it now with

```
./configure
make
make install
```

doesn't work the apple - now. so i have to switch on the pc later. and the thing - it is part of programming. like the algorithms. but know have to the algorithms, like insertion sort and selection sort and bubble sort.

so now we go on with the classical algorithms like:

- selection sort
- insertion sort
- bubble sort
- shell sort

and what is very import are algorithms in graphs. the thing there is deapth first search and the deapt the B - wide i don't know the word. but this is part of fernuni hagen - algorithmic math - the depth first search and in the width - ok. but - these algorithms for graphs - like in robert Sedgewick. Algorithms in C. and the next thing is: if we deterministic or not deterministic automat - we have to use graphs - and there is a algorithm, like deaepth first search. ok. and these algorithms use heaps, stacks, queues. and the python3 function method

```
Q.Top
Q.Append
Push
Pop
```

and so on. but these is part of C in this case, like Robert Sedgewick, Algorithm in C. so, it is necessary - to use. the algorithms in Graphs, but it is neccessary to learn the algorithm in python3 like algorithms mathematics, and like Robert Sedgewick. ok, now we start, but we start with inseration sort and selection sort. ok, no trouble we go smoking cigarettes in the morning

1 sorting algorithms

1.1 selection sort

```
int N = strlen (a);
int i, j;
int min;
int min_t;
int v_tmp;

for (i = 0;  i < N; i++) {
        min = i;
        for (j = i + 1;  j < N:  j++) {
                if (a [min] > a [j]) {
                        min_t = j;
                        j = min;
                        min = min_t;
                }
        }
        v_tmp = a [i];
        a [i] = a [min_t];
        a [min_t] = v_tmp;
}

// now we are testing
```

so gcc is part of the problem. and we have to learn

```
make gcc as ar ld ....
```

we take this code - snippet - and put head markers on the head - make a c file. and translate it with gcc. this works on apple iBook without

```
brew
port
```

and it works with

```
xcode-install
```

or sth. like that. we start.

The copyright on the head is not hard. so we take it like the police - i think everybody can use a (C) a copyright - not of use it for him/her/it self but, you can not put it out your room so easy. ok we go on. you don't have to think about such thinks!

```
// now we are testing

// (c) david vajda
// 11/09/2026
// insertion sort

// garden fance we call in german
// we don't need to explain our self
// why it is in german garden fance
// and we speak it now - no problem ...
// but it is neccessary to keep it in the haed - our head
// what/who ever head

#include <stdio.h>
#include <string.h>
#include <stdlib.h>

/*
int selectionsort (char *a) {
        int N = strlen (a);
        int i, j;
        int min;
        int min_t;
        int v_tmp;

        for (i = 0;  i < N; i++) {
                min = i;
                for (j = i + 1;  j < N:  j++) {
                        if (a [min] > a [j]) {
                                min_t = j;
                                j = min;
                                min = min_t;
                        }
                }
                v_tmp = a [i];
                a [i] = a [min_t];
                a [min_t] = v_tmp;
        }
return 0;
}
*/
/* so i have detected an error - we
have to swap/change/ the min
we don't have to swap - this is
part of the working process in laboratory
i wanted the message to shout out, to
see the error - the fault - the
and don't delete it not not show it, to
say here it doesn't work. so we don't excchance
the min indeces - swap - like the values.*/
```

```
// we comment out the wrong code

int selectionsort (char *a) {
        int N = strlen (a);
        int i, j;
        int min;
        int v_tmp;

        for (i = 0;  i < N; i++) {
                min = i;
                for (j = i + 1;  j < N;  j++) {
                        if (a [min] > a [j])
                                min = j;
                }
                v_tmp = a [i];
                a [i] = a [min];
                a [min] = v_tmp;
        }
return 0;
}

int main (void) {
        char a [] = "HASDJASDBASDBJASDASJDASD";
        printf ("%s\n", a);
        selectionsort (a);
        printf ("%s\n", a);
        printf ("good bye\n");
return 0;
}


/* yes this has worked.- very fine!!

davidvajda@localhost ~ % cp tmp11092026.tex insertionsort11092026.c
davidvajda@localhost ~ % gcc insertionsort11092026.c -o insertionsort11092026
davidvajda@localhost ~ % ./insertionsort11092026
HASDJASDBASDBJASDASJDASD
AAAAAABBDDDDDDHJJJSSSSSS
good bye
davidvajda@localhost ~ %

*/
```

ok. next thing is the insertion sort. it is the same - maybe - i have to read the book again, like selection sort. you have sth. the problem - i from 25 year know - but it has gone. not to find the difference, betweens these to algorithms. it are like the same, and if you are programming, you always tell your self the question. should I use this or that. and the thing is, these are the name - selection and insertion. we can make the swap - while sorting. than it is. insertion.

2 insertion

```
// (c) david vajda
// 11/09/2026
// insertion sort

#include <stdio.h>
#include <string.h>
#include <stdlib.h>


int insertionsort (char *a) {
        int N = strlen (a);
        int i, j;
        int min;
        int v_tmp;

        for (i = 0;  i < N; i++) {
                min = i;
                for (j = i + 1;  j < N;  j++) {
                        if (a [min] > a [j])
                                min = j;
                }
                v_tmp = a [i];
                a [i] = a [min];
                a [min] = v_tmp;
        }
return 0;
}

int main (void) {
        char a [] = "HASDJASDBASDBJASDASJDASD";
        printf ("%s\n", a);
        insertionsort (a);
        printf ("%s\n", a);
        printf ("good bye\n");
return 0;
}
```

sorry we took the right function name for the first algorithm, but we make it wrong on the title. cause also the filename was wrong, i changed it. and it is part of the protocol data/ i don't now in the bash and i put the whole storry in the bash here

```
davidvajda@localhost ~ % cp tmp11092026.tex insertionsort11092026.c
davidvajda@localhost ~ % gcc insertionsort11092026.c -o insertionsort11092026
davidvajda@localhost ~ % ./insertionsort11092026
HASDJASDBASDBJASDASJDASD
AAAAAABBDDDDDDDHJJJSSSSSS
good bye
davidvajda@localhost ~ % cp insertionsort11092026.c selectionsort11092026.c
```

```
davidvajda@localhost ~ % cp tmp11092026I.tex insertionsort11092026.c
davidvajda@localhost ~ % gcc insertionsort11092026.c -o insertionsort11092026
davidvajda@localhost ~ % ./insertionsort11092026
HASDJASDBASDBJASDASJDASD
AAAAAABBDDDDDDDHJJJSSSSSS
good bye
davidvajda@localhost ~ %
```

3 Knuth-Morris-Pratt

```
while ((p [i] != p [j]) && (j > 0))...
```

this we should keep in mind. an we should keep in mind

```
next [j] = j;
j = next [j];
...
```

like

```
while ((a [i] != p [j]) && (j > 0))...;
```

ok, we start

```
int i, j;
int N = strlen (a);
int M = strlen (p);

for (i = 0, j = 0;  (j < M) && (i < N);  i++, j = next [j]) {
        while ((p [i] != p [j]) && (j > 0))

}

// we have to find the difference - by pre running
// by init - what ever. so in the other case
// when it is the pre run/what ever
// we have to: set j to -1. and we can find
// the things which are equl from the one code to the other
// but
// again

int i, j;
int N = strlen (a);
int M = strlen (p);

// this is in the searching process
// the j is equal to next [j]
// and the pre run - there is: this is the searching process? i don't know
// wait a little

for (i = 0, j = 0;  (j < M) && (i < N);  i++, j = next [j]) {
```

```
        while ((p [i] != p [j]) && (j > 0))
                next [j] = j;

}

for (i = 0, j = -1;  (j < M) && (i < N);  i++, j++) {
        while ((a [i] != p [j]) && (j > 0))
                j = next [j];
}

// the question is: no problem i have it in a book, is in this
// part - j++ in the outer loop? this is the question
// when it is not equal in the inner loop
// a [i] != p [j]
// than: j = next [j]
// this is ok. and: but this is the inner loop.
// there is a j = -1.

// ok. no problem. we write, the strict - i don't know
// the name at the moment - unprocess/unworked i don't know
// name - raw. i don't know
```

but this works others. here is the is not equal and is equal loop in the inner loop
so, go on

```
for (i = 0;  j = 0;  (j < M) && (i < N);  i++, j = 0)
        while ((a [i] == p [j]) && (j < M)) {
                j++;
                i++;
        }
```

we could use this. but, go on. so, we have to unterscheiden, between searching
the pattern in the pattern. and searching the pattern in the text.
we look in the book

```
int kmpsearch (char *p, char *a) {
        int i, j, M = strlen (p), N = strlen (a);
        initnext (p);
        for (i = 0, j = 0; (j < M) && (i < N);  i++, j++) {
                while ((j >= 0) && (a [i] != p [j]))
                        j = next [j];
        }
        if ...
}
```

this maybe is ok. but this is not initnext

```
initnext (char *p) {
        int i, j, M = strlen (p);
        next [0] = -1;
        for (i = 0, j = -1;  i < M;  i++, j++, next [i] = j)
                while ((j >= ) && (p[i] != p [j]))
```

```
                                    j = next [j];
}
```

so ok.

4 quine mc cluskey

```
(C) David Vajda
Mon Nov  9 09:38:58 2026
4 Network - TTL - Disjunktive Normalform

 0  0  0  0  0      0
 1  0  0  0  1      0
 2  0  0  1  0      0
 3  0  0  1  1      1
 4  0  1  0  0      1
 5  0  1  0  1      0
 6  0  1  1  0      1
 7  0  1  1  1      1
 8  1  0  0  0      1
 9  1  0  0  1      1
10  1  0  1  0      1
11  1  0  1  1      0
12  1  1  0  0      1
13  1  1  0  1      0
14  1  1  1  0      1
15  1  1  1  1      1
```

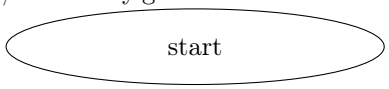
so, now the table in LaTeXVersion, not like lstlisting

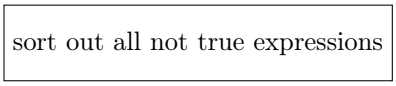
(C) David Vajda
Mon Nov 9 09:38:58 2026
4 Network - TTL - Disjunktive Normalform

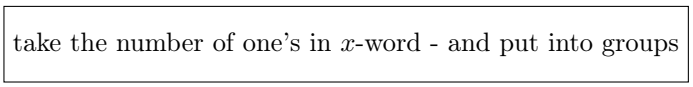
n	x_3	x_2	x_1	x_0	y
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	1
15	1	1	1	1	1

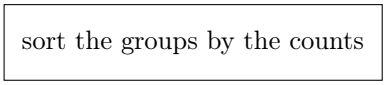
so, now the typically - übliche - al-

gorithm - for minimate - reduce - the expressions. in quine.
this is an example, which doesn/t look very good - i show the source. and if this is



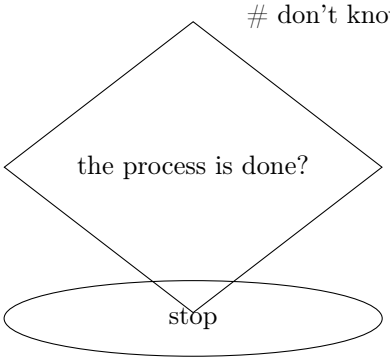








don't know if process are forgotten here - this is a fra



ready.

```
\begin{tikzpicture}
\node [ellipse ,draw=black ,minimum width=5cm, minimum height=1cm] (start) at (2.

\node [rectangle ,draw=black ,minimum width=5cm, minimum height=1cm] (stmt1) at (

\node [rectangle ,draw=black ,minimum width=5cm, minimum height=1cm] (stmt2) at (

\node [rectangle ,draw=black ,minimum width=5cm, minimum height=1cm] (stmt3) at (

\node [rectangle ,draw=black ,minimum width=5cm, minimum height=1cm] (stmt4) at (
```

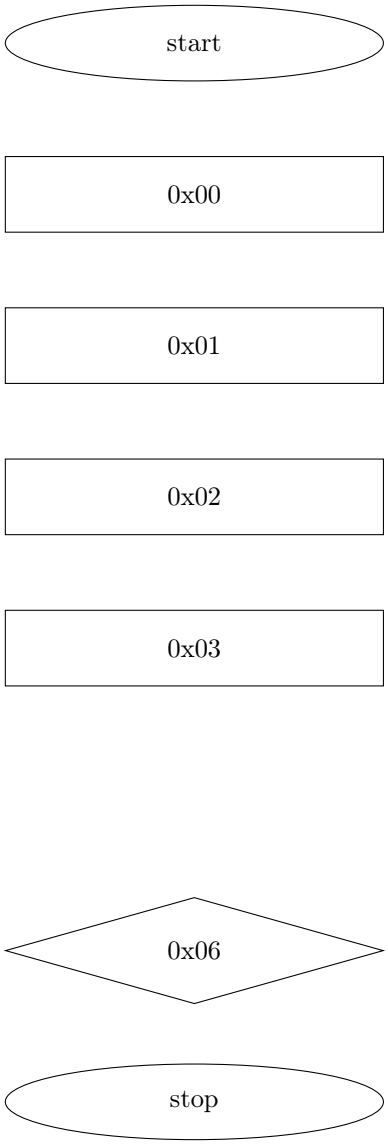
```
\node [rectangle,minimum width=5cm, minimum height=1cm] (comment1) at (8,6) {\#

\node [rectangle,minimum width=5cm, minimum height=1cm] (comment2) at (8,4) {\#

\node [diamond,draw=black,minimum width=5cm, minimum height=1cm] (stmtcond5) at

\node [ellipse,draw=black,minimum width=5cm, minimum height=1cm] (stop) at (2.5
\end{tikzpicture}
```

the difference, between this and that is, following: we have - Condition A, this is not a condition, better a description, or what ever. no condition. This is Like A000 0001. or what ever. so we number the states and put the description anywhere else



now:

0x00: sort out all not true expressions

```
0x01: take the number of one's in x-word - and put into groups
0x02: sort the groups by the counts
0x03 :verschmelzen (?) (not merge) the expression with i ones and i+1
0x04: there is one input variable x less, and this is like three cables}
0x05: don't know if process are forgotten here - this is a fragment
0x06: the process is done?
```

ok, now we go on. so. we do quine mc cluskey.

(C) David Vajda
Mon Nov 9 09:38:58 2026
4 Network - TTL - Disjunktive Normalform

n	x_3	x_2	x_1	x_0	y
3	0	0	1	1	1
4	0	1	0	0	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	1
12	1	1	0	0	1
14	1	1	1	0	1
15	1	1	1	1	1

we put it into an lstlisting - because, the table in LaTeXis hard to read.

```
\begin{tabular}{|c|c|c|c|c|}
\hline
\multicolumn{6}{|l|}{\small (C) David Vajda}\\
\multicolumn{6}{|l|}{\small Mon Nov  9 09:38:58 2026}\\
\multicolumn{6}{|l|}{\small 4 Network - TTL - Disjunktive Normalform}\\
$n$ & $x_3$ & $x_2$ & $x_1$ & $x_0$ & $y$\\
\hline
 $0$ & $0$ & $0$ & $0$ & $0$ & $0$\\
 $1$ & $0$ & $0$ & $0$ & $1$ & $0$\\
 $2$ & $0$ & $0$ & $1$ & $0$ & $0$\\
 $3$ & $0$ & $0$ & $1$ & $1$ & $1$\\
 $4$ & $0$ & $1$ & $0$ & $0$ & $1$\\
 $5$ & $0$ & $1$ & $0$ & $1$ & $0$\\
 $6$ & $0$ & $1$ & $1$ & $0$ & $1$\\
 $7$ & $0$ & $1$ & $1$ & $1$ & $1$\\
 $8$ & $1$ & $0$ & $0$ & $0$ & $1$\\
 $9$ & $1$ & $0$ & $0$ & $1$ & $1$\\
$10$ & $1$ & $0$ & $1$ & $0$ & $1$\\
$11$ & $1$ & $0$ & $1$ & $1$ & $0$\\
$12$ & $1$ & $1$ & $0$ & $0$ & $1$\\
$13$ & $1$ & $1$ & $0$ & $1$ & $0$\\
$14$ & $1$ & $1$ & $1$ & $0$ & $1$\\
$15$ & $1$ & $1$ & $1$ & $1$ & $1$\\
\hline
\end{tabular}
```

(C) David Vajda

```
Mon Nov  9 09:38:58 2026
4 Network - TTL - Disjunktive Normalform

 3  0  0  1  1     1
 4  0  1  0  0     1
 6  0  1  1  0     1
 7  0  1  1  1     1
 8  1  0  0  0     1
 9  1  0  0  1     1
10  1  0  1  0     1
12  1  1  0  0     1
14  1  1  1  0     1
15  1  1  1  1     1


Group 1:
   4  0  1  0  0     1
   8  1  0  0  0     1
Group 2:
   3  0  0  1  1     1
   6  0  1  1  0     1
   9  1  0  0  1     1
10  1  0  1  0     1
12  1  1  0  0     1
Group 3:
    7  0  1  1  1     1
 14  1  1  1  0     1
Group 4:
 15  1  1  1  1     1

(4,6)             0 1 x 0
(4,12)   x 1 0 0
(8,9)             1 0 0 x
(8,10)   1 0 x 0
(8,12)   1 x 0 0
(3,7)             0 x 1 1
(6,7)             0 1 1 x
(6,14)   x 1 1 0
(10,14)  1 x 1 0
(12,14)  1 1 x 0
(7,15)   x 1 1 1
(14,15)  1 1 1 x

# go on!

a step sorting groups by cables - pins - bits at the same position
```

```
(4,12)  x 1 0 0
(6,14)  x 1 1 0
(7,15)  x 1 1 1

(8,9)           1 0 0 x
(14,15) 1 1 1 x
(6,7)           0 1 1 x

(8,10)  1 0 x 0
(12,14) 1 1 x 0
(4,6)           0 1 x 0

(8,12)  1 x 0 0
(3,7)           0 x 1 1
(10,14) 1 x 1 0

# so go on! we - verschmelze them.
# before that, we give them, the group
# numbers

Group 1 =
(4,12)  x 1 0 0
Group 2 =
(6,14)  x 1 1 0
Group 3 =
(7,15)  x 1 1 1

Group 1
(8,9)           1 0 0 x
(14,15) 1 1 1 x ## there is a
## step/process/pipeline... missing
## we have to sort them, or this is
## the second thing
(6,7)           0 1 1 x

(8,10)  1 0 x 0
(12,14) 1 1 x 0
(4,6)           0 1 x 0

(8,12)  1 x 0 0
(3,7)           0 x 1 1
(10,14) 1 x 1 0

again:

Group 1 =
(4,12)  x 1 0 0
Group 2 =
(6,14)  x 1 1 0
Group 3 =
```

```
(7,15)   x 1 1 1

Group 1 =
(8,9)              1 0 0 x
Group 3 =
(14,15) 1 1 1 x
Group 2 =
(6,7)              0 1 1 x

Group 1 =
(8,10)   1 0 x 0
(4,6)              0 1 x 0
Group 2 =
(12,14) 1 1 x 0

Group 1 =
(8,12)   1 x 0 0
Group 2 =
(3,7)              0 x 1 1
(10,14) 1 x 1 0

now sorted:


Group 1 =
(4,12)   x 1 0 0
Group 2 =
(6,14)   x 1 1 0
Group 3 =
(7,15)   x 1 1 1

Group 1 =
(8,9)              1 0 0 x
Group 2 =
(6,7)              0 1 1 x
Group 3 =
(14,15) 1 1 1 x


Group 1 =
(8,10)   1 0 x 0
(4,6)              0 1 x 0
Group 2 =
(12,14) 1 1 x 0

Group 1 =
(8,12)   1 x 0 0
Group 2 =
(3,7)              0 x 1 1
(10,14) 1 x 1 0
```

```
and now we are - bringing them together.


Group 1 =
(4,12)  x 1 0 0
Group 2 =
(6,14)  x 1 1 0
Group 3 =
(7,15)  x 1 1 1

(4,12,6,14)     x 1 x 0
(6,14,7,15)     x 1 1 x

Group 1 =
(8,9)           1 0 0 x
Group 2 =
(6,7)           0 1 1 x
Group 3 =
(14,15) 1 1 1 x

this doen't work with - verschmelzen
this will be like before
        (8,9)           1 0 0 x

(6,7,14,15)             x 1 1 x


Group 1 =
(8,10)  1 0 x 0
(4,6)           0 1 x 0
Group 2 =
(12,14) 1 1 x 0

(8,10,12,14)            1 x x 0
(4,6,12,14)             x 1 x 0

Group 1 =
(8,12)  1 x 0 0
Group 2 =
(3,7)           0 x 1 1
(10,14) 1 x 1 0

(8,12,10,14)            1 x x 0
(3,7)                           0 x 1 1

we filter out the unneccessary things

(4,12,6,14)             x 1 x 0
(6,14,7,15)             x 1 1 x
```

```
(6,7,14,15)          x 1 1 x
(8,10,12,14)         1 x x 0
(4,6,12,14)          x 1 x 0
(8,12,10,14)         1 x x 0
(3,7)                      0 x 1 1
```

now we take out the double and more than one's

```
(4,12,6,14)          x 1 x 0
(6,14,7,15)          x 1 1 x
(8,10,12,14)         1 x x 0
(3,7)                      0 x 1 1
```

so, maybe this will be finished, when x are at the same position.
and we can make minimal rest – ueberdeckung.

	3	4	6	7	8	10	12
(4,12,6,14)	1	1				1	1
(6,14,7,15)		1	1				1
(8,10,12,14)				1	1	1	1
(3,7)	1			1			

these are all primimplikanten

	3	4	6	7	8	10	12	14	15
(4,12,6,14)		1	1				1	1	
(6,14,7,15)			1	1				1	1
(8,10,12,14)					1	1	1	1	
(3,7)	1			1					

and now we can make the forumula.

```
(4,12,6,14)          x 1 x 0
(6,14,7,15)          x 1 1 x
(8,10,12,14)         1 x x 0
(3,7)                      0 x 1 1
```

$$
\begin{aligned}
y &= x_2 \wedge \neg x_0 && (1)\\
&\vee\ x_2 \wedge x_1 && (2)\\
&\vee\ x_3 \wedge \neg x_0 && (3)\\
&\vee\ \neg x_3 \wedge x_1 \wedge x_0 && (4)\\
& && (5)
\end{aligned}
$$

ok, now this formula - with galasm

```
gal16v8
module gal16v8dnf11092026

clk x3 x2 x1 x0 nc nc nc nc gnd
```

/oe y nc nc nc nc nc nc nc vcc

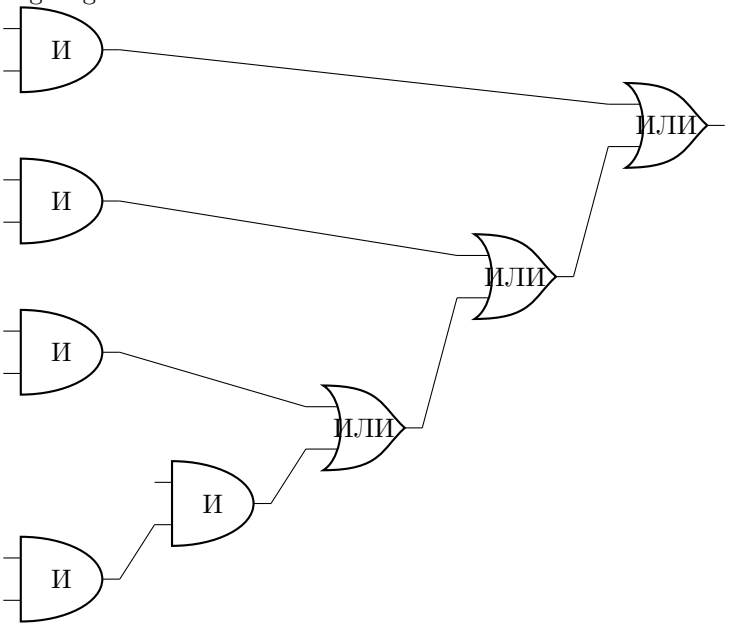
y = x2*/x0 + x2*x1 + x3*/x0 + /x3*x1*x0

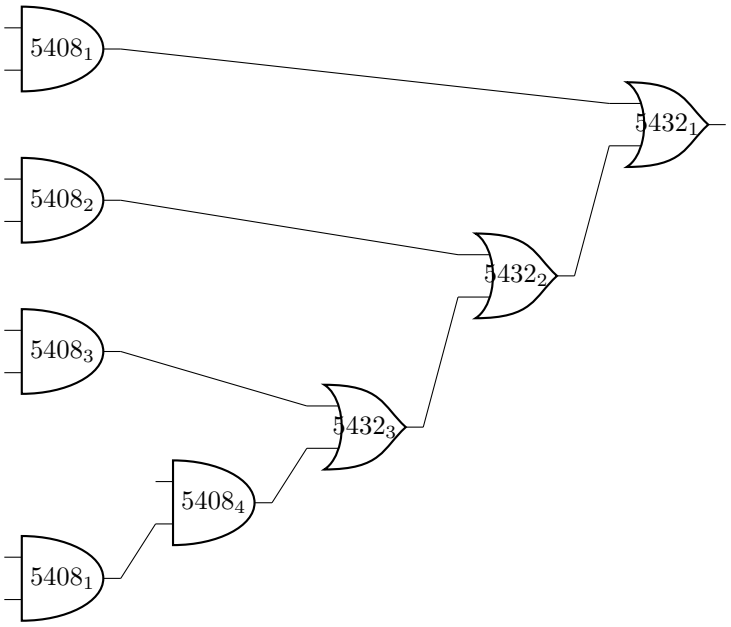
description: the hole day formula

and now: with tikzpicture

we have: 5 and gates.

so, sorry for today - russian and cyrrillic doesn't work right now in this document. it is a small and short and long storry - to learn the input character. it is maybe easy - i have a LaTeXBook. and a: UTF-8, Aiken, Dreiexcess, ISO-8859-1 usw. but in this case i don't know. input character part of learning - now i don't know. so doesn't work, we go on make. make the text ready for homepage and Facebook. and. go back to programming and go back to UML - cause, we want to make the Algorithms in C - we want to make a class library. and: we implement lineare Kongruenz, additive and so on and so on. and: classes, so sortings algorithms all



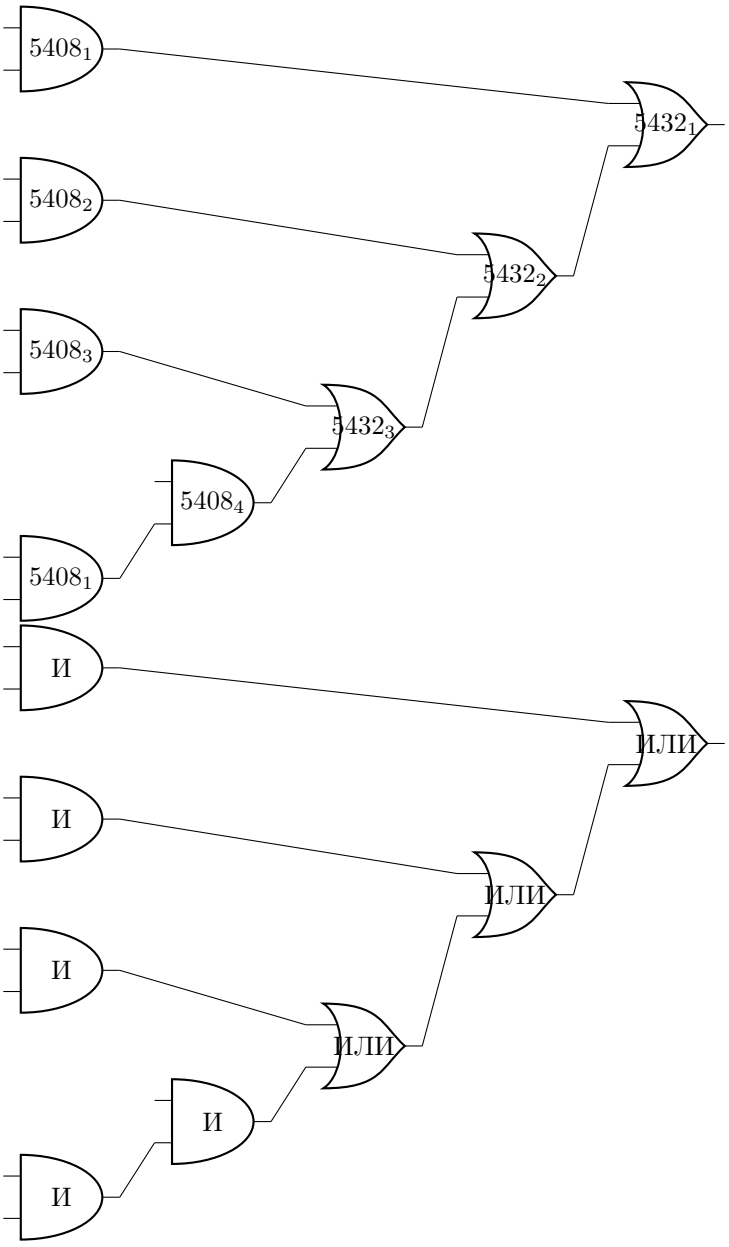


and again: and now we have to put in the not gates in russian - ne, in russia cyrillic: . i can't put directly the UTF-8 Code arabian in the LaTeX. and the slash command, i think doesn't these characters. i go on. i put the code of the last tikzpicture here. and i will make a new with - not gates.

```
\begin{tikzpicture}
\node [and port] (and1) at (0,7) {$5408_1$};
\node [and port] (and2) at (0,5) {$5408_2$};
\node [and port] (and3) at (0,3) {$5408_3$};
\node [and port] (and4) at (2,1) {$5408_4$};
\node [and port] (and5) at (0,0) {$5408_1$};

\node [or port] (or1) at (8,6) {$5432_1$};
\node [or port] (or2) at (6,4) {$5432_2$};
\node [or port] (or3) at (4,2) {$5432_3$};

\draw (or3.out) -- (or2.in 2);
\draw (or2.out) -- (or1.in 2);
\draw (and5.out) -- (and4.in 2);
\draw (and4.out) -- (or3.in 2);
\draw (and3.out) -- (or3.in 1);
\draw (and2.out) -- (or2.in 1);
\draw (and1.out) -- (or1.in 1);
\end{tikzpicture}
```



5 an message error again

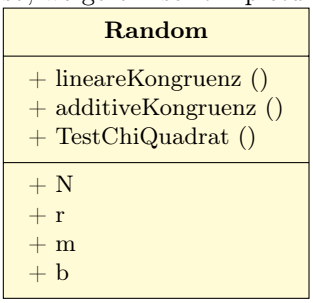
it had worked - i didn't know - the problem a forgotten brace. and now it works with cyrillic character. the thing is, the package has to be bind in is:

\usepackage[T2A]{fontenc}

with this it works, like in the other - but - there was a forgotten brace. so. and now if/when/then this works in LaTeX. with russia/cyrillic, we can go on and do the next steps before we put it into the homepage and fine in facebook, with

use of the PC. the not gates will follow - this is the first thing, and the second is. UML and: we do: the algorithms in C. and so on.

so, we go on. so: tikzpicture comes now - but with UML



ok, this is, what I had the last time. the UML Class in LaTeXis not hard 2 use!

```
\begin{tikzpicture} \umlclass{Random}{
+ lineareKongruenz ()\\
+ additiveKongruenz ()\\
+ TestChiQuadrat ()}{+ N\\
+ r\\
+ m\\
+ b}\\
\end{tikzpicture}
```

should we make a logical expression - the word - consumption sounds in my ears. should we make a logical consumption. what is this for a word. marx tells: the konsumerable ... means the eating and nibbling at meal times. because a good worker - needs - how we say in english so much time to be - ausgebildet - ex.... there are costs for heatings and eatings. and we have an beginning capital. so an logical consumption. i don't know what it means. but I read a book - no not a book - a - bericht about an gangster in the www - the press. the press said - a gangster (A, B, C). Logical expressions. i think this is not always not forbitten not to do this, because other people are having trouble. i don't think so, it are good examples. the gangster his nervs are finished, this is, what we have to do with other people, without it is, aggressiv and against material things.

we go on. in another context I found out, the logical expressions:

A: he is a rubber
B: i don't know

C: i can go to the judge

$$A \wedge \neg B \quad \Rightarrow \quad C \tag{6}$$

(7)

ok. this is nice. because:

$$C \quad \Rightarrow \quad \neg(A \wedge \neg B) \tag{8}$$

(9)

$$C \Rightarrow \neg A \vee B) \tag{10}$$
$$\tag{11}$$

ok, i go to the judge meens he is no gangster and don't know it? ok, we can do it in examples. also, but i found out, that:

A: we make an excersize every day

i don't know now - what excactly i wanted to say

B: i know what i want to say

here it is not necessary. but i think, we could make it in the chemistry. $A0001$ i don't know any more, what i wanted to say, maybe it comes back, later. ok. what i wanted to do, is, that LaTeXUML is easy to use, at this point.

```
\begin{tikzpicture} \umlclass{Random}{
+ lineareKongruenz ()\\
+ additiveKongruenz ()\\
+ TestChiQuadrat ()}{+ N\\
+ r\\
+ m\\
+ b}\\
\end{tikzpicture}
```

there is like always -

```
\umlclass
\end{listlisting}
this is not hard to use. we have, easy.
\begin{lstlisting}
\begin{tikzpicture}
```

we had now often used.

```
\begin{tikzpicture}
\node [and port] (and1) at (0,0) {text};
```

the UML Class

```
\umlclass
```

is even very easy to use. in braces - the - nice rackts/brackets, what ever, comes, the name of the class, 2 will follow the methods and the attributs. ok, until now. than will follow, like always, and the codes. we put into to a text file. we put - in this case the relationships to this - the entites - and what we are doing: the mathematical expressions. LaTeXMath Style with dollars, like pounds and euros. now!

so we concentrate now on the aufgabe. to find out all - methods functions we want to integrate. and first make a list

```
methode der lineare Kongruenz (Zufallszahlen)
method of additiv congruency (RandomNumbers)
Chi Square Test
```

kurvenanpassung/kurvenglaettung in english
Lagrange – interpolation
splines
method of smallest/minimum squares

then: integration:
symbolic
elementary squaremethod (quadraturverfahren)
zusammengesetz – joined – i don't square – quadratur – i don't know
adaptive method

verfahren: method?

than: FFT – as one thing – it is with: einheitswurzeln – complex, i don't know

ok: than: comes
breath first search
depth first search
eulertour and hamilton

and than comes:
KMP – Knuth Morris Pratt
Boyer Moore
Rabin Karp
Brute Force – this is the right word – 4 grob

than: sorting methods:
insertion sort
selection sort
bubble sort
shell sort
quicksort

and than:
nichtdeterministischer suchautomat

compiler,
top–down–parsing
bottom–up

and: ok, and we make as excersize – we make – it first, with. the names
in real C++ or what ever, like programming

hash tables are missing
and what is neccessary:

with mathemcals class expressions. and the UML Class – diagramms with
Mathematics, like LaTeX Math Expression. First, we give the formula for the for
i don't know how to express that and we start. also:

```
erwartungswert
varianz
standardabweichung
and so on

we have a plain text list here. and we make an itemize list. but first, some m
formula \LaTeX Formulas.
```

Lagrange:

$$l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \tag{12}$$

$$\tag{13}$$

in the next step we make the lagrange formula real:

$$p(x) = \sum_{i=0}^{n} y \cdot l_i(x) \tag{14}$$

$$\tag{15}$$

i think. maybe the last consists by/from errors. i don't. we have to look in the book. in wikipedia. and the codes, we find in the book algorithm in C. this is mathematical expression.

now follows the folge - i don't know the english word. with: linaery congruency - i think - and additive. it is

$$(a_i)$$

an Alpha not subset. because. the set is the right word for list. in the other way. which is an. relation by products of sets. and an database - relationship Database what ever. and then now

$$(a_i) = (a_{i-1} \cdot b + 1) \mod m \tag{16}$$

$$\tag{17}$$

and the joke - what ever is - in german joke, is: in hash tables, the crucial - is this the right word? we need: v or what ever, instead of the always $+1$. and we could think of additve here are two such random numbers, lists, but, this is not the hole true. a little bit like - KMP - where we search in the pattern - and the rules are nearly the same, we take the beginning of the list of random numbers, to make this additiv. but it is more or ... (english) like - hash table and random numbers - and a little like - KMP. but there must be - a rule. which is mathematicaly. cause knuth found out - that m must 2^{32} maybe. and b must one digit less. must ends on 21. must be odd i think even is $2, 4, 6, \ldots$
ok now. we have 2 or three expressions and write. the classes. in UML, what is not complicate.
words, expressions, vocabulary maybe are:

- curve fitting
- linear curve fit
- regression/least square fitting - says google AI - [1]least square maybe is the right word.
- interpolation

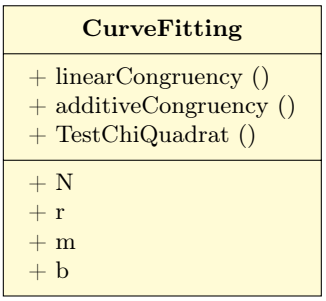
least square fitting: methode der kleinsten quadrate. and this could be reggression. i don't know what reggression and had no time to learn brutto and netto. but i know

$$quotient = \frac{dividend}{divisor} + rest \tag{18}$$

$$\tag{19}$$

so it is possible - and possible to learn what reggression really is. maybe it is - least square fitting - like alway - the x.000.000.000 Text like the - and the smiling like ever. but the difference. the need is clear. in economics. there is now need - to go to the bucher - and make a deal. sorry. and the number x.000.000.000 little different. the is a small circle in big circle.
ok - go on!

so:



we forgot the last. The Test.

$$\chi^2 = \sum \frac{\left(f_i - \frac{N}{r}\right)^2}{\frac{N}{r}} \tag{20}$$

$$\tag{21}$$

now the class in UML, with mathematics. but this was random. a beginners class.

CurveFittingAndRandom
+ $(a_i) = (a_{i-1} \cdot b + 1) \mod m$ + additiveCongruency () + $\chi^2 = \sum \frac{\left(f_i - \frac{N}{r}\right)^2}{\frac{N}{r}}$ $l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}$ $p(x) = \sum_{i=0}^{n} y \cdot l_i(x)$
+ N + r + m + b + $\{x_0, x_1, \ldots, x_n\}$ + $\{y_0, y_1, \ldots, y_n\}$ + i + j

now a next class is: the

- erwartungswert
- standardabweichung
- varianz

what it means in english?

- erwartungswert: Expected value
- standardabweichung: Standard deviation
- varianz: Variance

this is not roh like - a roh - this is a small sigma like

$$\sigma$$

and this in german small s. and s square. ok.

- erwartungswert: Expected value
- standardabweichung: Standard deviation σ
- varianz: Variance σ^2

RealRandom
$X(x_i) = f(x_i)$ $E(x) = \sum_{i=0}^{n} p(X = x_i) \cdot X(x_i)$ $\sigma = \sum_{i=0}^{n} (x_i - E(X))^2$ the fantasy formula - without p(x) have to learn $Var(\sigma^2) = \sqrt{(\sigma)}$ $\binom{n}{k} = \frac{k!(n-k)!}{k!}$ fantasia formula $h(E) = \frac{H}{n}$ $P(X)$
$S_{dice} = \{1, 2, 3, 4, 5, 6\}$ $S_1 = \{r, b\}$ $S_{coin} = \{wappen, number\}$ $S = \{x_0, x_1, x_2, \ldots, x_n\}$ n H

and the class is

6 next step

implementing in code. try to remember. what are missing in the case david vajda - born 1983 - tuebingen - coffee exception:

```
try {
        bla, bla, bla;
        bla, bla, bla;
        bla ...
}
catch (CoffeException) {
        must: run (hard: systemcall) signal: hangup, drinkcoffee
}
```

the vajda - (breite, laenge) grad - tuebingen, 1983 n. Chr. so, we try to - find - in head we try to remember - the KMP algorithm. Knuth-Morris-Pratt. and we try to - find out - the Depth First Search - and Breath First Search. The first Algorithm I have in python3 and other pythons. and it is part of Algorithms in C - so in C - but have to take a serious look on the topic. cause, the stack, the queue and the heap - make problems in backround of the brain - exactly - at his point - liks (breite,laenge) tuebingen. ok we go on.
frist KMP

```
kmpsearch (char *a, char *p) {
        int i;
        int j;
        int N = strlen (a);
        int M = strlen (p);

        kmpinit (p);
```

```
        for (i = 0, j = -1; (i < N) && (j < M);  i++, j++) {
                while ((a[i] != p[j]) && (j > 0))
                        j = next[j];
        }
        //... missing code
}

kmpinit (char *p) {
        int i;
        int j;
        int M = strlen (p);

        for (i = 0, j = -1; (i < N) && (j < M);  i++, next[j] = j) {
                while ((p[i] != p[j]) && (j > 0))
                        j = next[j];
        }
        //...
}
```

better I can't remember. ok. next. the next are the to sortings algorithms, insertion sort, and selection sort. we make it fast.

```
selectionsort (char *a) {
        int i, j;
        int min;
        int t;

        int N = strlen (a);

        for (i = 0; i < N; i++) {
                min = i;
                for (j = i + 1;  j < N;  j++)
                        if (a[min] > a[j])
                                min = j;
                t = a[min];
                a[min] = a[i];
                a[i] = t;
        }
}
```

this is selection sort and I will not do the insertion sort. now I try the thing - the breath first search! so. we try to have a look in this code. it comes from algorithmische mathematik, fernuni hagen, 61411, Hochstädtler, Schulte, Nickel, Andres.

```
pred[r] = r
component[r] = r
Q.Append(r)
while Q.IsNotEmpty():
        v = Q.Top()
        for w in Neighborhood(v):
```

```
            if pred[w] == None:
                pred[w] = v
                component[w] = component[v]
                Q.Append(w)
```

maybe the code doesn't differ from the code algorithm in C, by robert sedgewick. and Robert Sedgewick has implemented. Queues, Heaps and Stack. He: has: the Word, Top (), always - and now we go on.

now we look at this: the only thing which makes it different, is: that:

```
component
pred
```

exisits. component will - this is the komponent in a graph, and subgraphs, but - a component, we can make our euler graph, exisiting, we: make a graph, where every node is a neighbour. and a node has all nodes as neighbours. if this is happend. we can make components, were all nodes has 2 vertices, 2 neighbours. and when all nodes has to neighbours, while that other can have 4 and 6 and so on.ok.

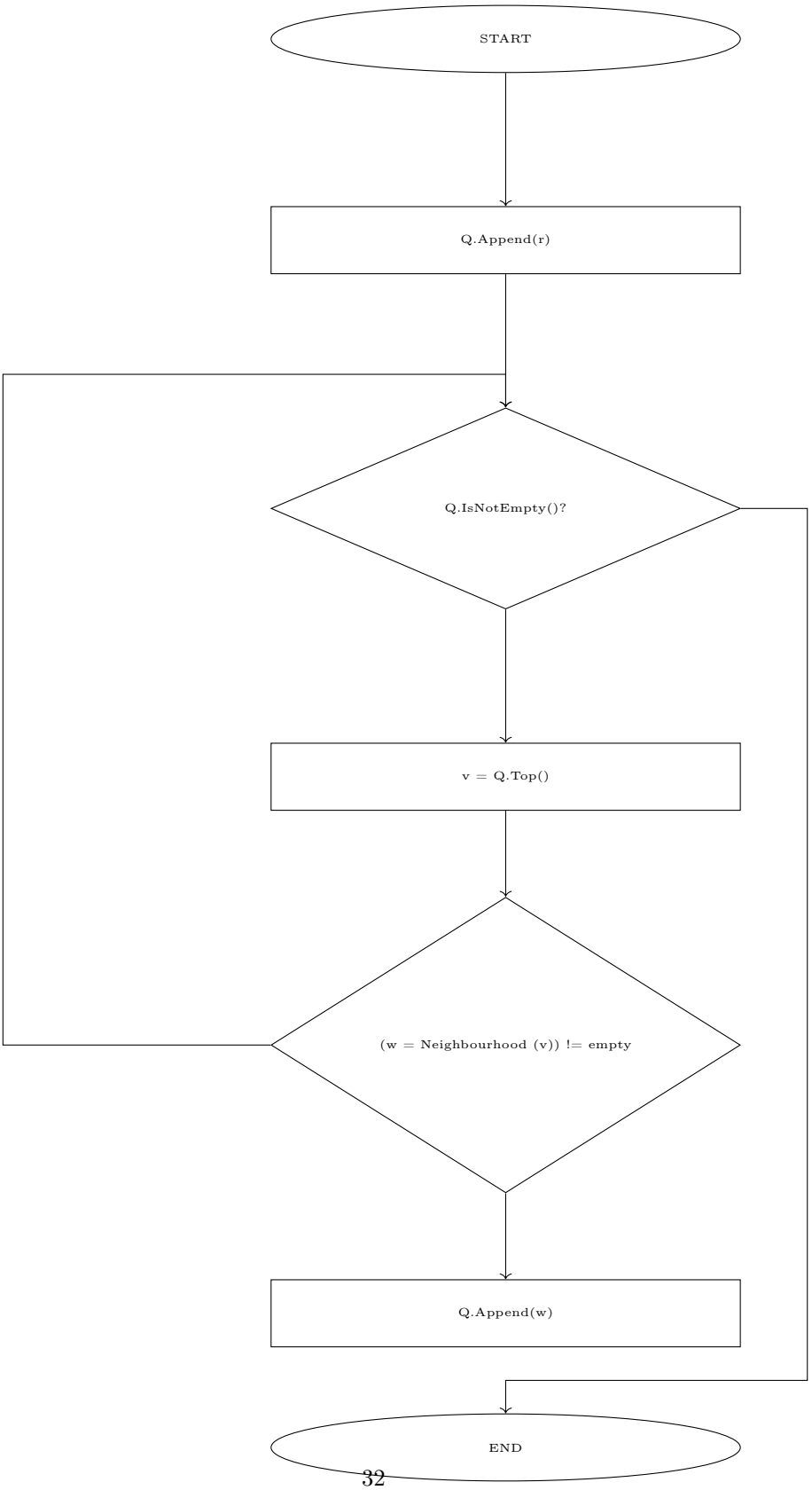
the Thing is:

```
Q.Append(r)
while Q.IsNotEmpty():
        v = Q.Top()
        for w in Neighborhood(v):
            if pred[w] == None:
                Q.Append(w)
```

the breath first search, without component, and without pred, but this doesn't really works. would look like this. explanation.

1. We put the r at the top of the Queue
2. While the Queue is not empty
   we can make a struktogramm/PAP out of these listings of Statements
3. We take the head of the Queue
4. we take all the neighbours of the head of the queue and we append it.

now we make a diagramm


START
Q.Append(r)
Q.IsNotEmpty()?
v = Q.Top()
(w = Neighbourhood (v)) != empty
Q.Append(w)
END

so, now we go on.

Sorry about yesterday. I know the ßimpleformula for the binomial coefficient by heart—once I'm up and about, I'll recite it for you!

```
(n choose k) = n! / (k! * (n–k)!)
```

I know that by heart!

```
(n choose k) = n! divided by k! times (n–k)!
```

And by the way:
We can also make use of Taylor polynomials.
And we're already working hard on trying out the Lagrange method. It would also be nice to look at some classical—oh, right, in English—more on that later...

$$\left(\begin{array}{c} n \\ k \end{array} \right) = \frac{n!}{k! \cdot (n-k)!} \tag{22}$$

$$\tag{23}$$

this is an easiest aufgabe, to make the calculation. we can. calcualty the faculty 4, for example. and with 2. and we can try. to calculate.

$$\begin{array}{c} 4 \\ 2 \end{array}$$

this we can do later. the maintanance. is not to abweichen, from the typicall content. the thing, we can expand the topics about:

- Faculty - no problems - and Binomialcoefficient
- but we are, in algorithms
- and we have to take serious the real hardware, like ARM und AVR (ATmega8 4 example). this means, we need to take the thinks in assembler languages serious

  UBBRH, UBRRL
  UCSRC, UCSRB, UCSRA

  this for: RS-232, v.24
- but, the: Lagrange, integration and curve fitting is maybe an example, but it is not a good idea, to make this the topic for the live, the only is to realise that in future. so we have to go on. the AD wandler. is not the first primary topic, it is but, because, of the architecture, of ATmega8, like of much architecture. the next thing, what we have to do, is
- the sorting algorithms: insertion sort
- selection sort
- bubble sort

- quick sort
- taylor - reihe - what is reihe? we need to watch - and folge
- and: but: fourier, is importand, and FFT, curve fitting
- important, is: lagrange - and this without curve fitting
- than: taylor/lagrange
- then comes: Breath First Seaerch, Depth First Search
- than comes: Euler.
- lagrange - taylor
- and we can make: interpolation with lagrange
- we can make: varianz, what was the name for: Erwartungswert and Variance.
- erwartungswert: Expected value
- standardabweichung: Standard deviation
- varianz: Variance
- these are things: we can do in the mathematical - expected value, standard deviation, variance.
- these are things, we can calculate easy
- the formula for: linear congruency, additive congruency. these are the two things.
- and we can do: taylor reihe? i have ask, google ai
- Taylorreihe: Taylor series
- Fourierreihe: Fourier series
- series is the Reihe. and what is the folge?
- here is written: Reihe (mathematisch): Series (Plural ebenfalls series)
- series and series. ok, no problem. I have found out in the electrical yellow book: there are two forms signal - a series and sth. else. we can keep this in mind
- the next thing is, what we have to do: we are: with: taylor and fourier and integration, are our topics, and the next things: are: integration and curve fitting, we can put this into UML diagramms and we have to use this in LaTeX
- this is a std. exersize. is: FFT and Integration. like. and the thing is: we have symbolic - we don't solve it with riemann, even if squares and methods are part of the theme ok.

- the other algorithms are important: like: Knuth Morris Pratt, Boyer Moore, Rabin Karp.
- but, the thing not to forget, the hardware - the: CPU - atmega8, AVR, ARM Cortex, maybe we can start. But we have to: take care, not to loose, the WDH. of the always same things
- Even: IEEE-1984/91 is a theme, with tikzpicture, circuitikz, and we have to do the quine, because it is, a hard excersise, even it doesn't look so, to: pin on the Gates, like AND OR NOT - nodes
- the PAP/ASM Diagramm has to be done - the other thing: the automat. has to be done. like always - mealy and moore. transformation, out of the mathematical calculation and so on
- so let's go on: the things are: we have also, to make every da - the atmega8, m8 codes. for LED Blinking, for using the Ports, RS-232, V.24, and so on. these are std. things and go on
- what we have to do: is: integration and fourier, Curve fitting, congruency, exptected value, standard deviation, variance and so on.
- and than: yeah, the depth first search and the breath first search and: the euler tour, all that, has to be done, and algorithm like. so let's go on

not start with quine! to day! some atmega8, arm Cortex specific - excersize - the PAP we can do later. The PAP comes later. so. we make. a interpolation lagrange. the first - implementation in mathematics - and after that, in the afternoon or what ever, the implementation in code. a city grow up, and we make houses or towers every but we make all towers, for the half, in the second working process - in the second half live of the town buildung - build the second half on the city. the ground inspection. let's go on. we make the ASM Code, AVR. Ok, litte python, what mathematical definition linear congruency, lagrange - what ever, and the formular for additive is easy

7 atmega8

7.1 m8 code - easiest - the first

```
;; (c) david vajda
;; atmega8/m8 codes for excersize
;; 11/10/2026

.include "m8def.inc"

ldi r16, HIGH (RAMEND)
out SPH, r16
ldi r16, LOW (RAMEND)
out SPL, r16

ldi r16, 0xff
```

```
out DDRB, r16

ldi r16, 0b10101010
out PORTB, r16

main: rjmp main
```

7.2 m8 code - and the count with waiting loop

```
;; (c) david vajda
;; atmega8/m8 codes for excersize
;; 11/10/2026

.include "m8def.inc"

ldi r16, HIGH (RAMEND)
out SPH, r16
ldi r16, LOW (RAMEND)
out SPL, r16

ldi r16, 0xff
out DDRB, r16

ldi r16, 0xff
main01:
out PORTB, r16
rcall waitingloop
dec r16
rjmp main01

main: rjmp main

waitingloop:
push r16
push r17
push r18
ldi r16, 0xff
waitloopl3:
ldi r17, 0xff
waitloopl2:
ldi r18, 0x04
waitloopl1:
dec r18
brnz r18, waitloopl1
;; maybe: brnz but brne -
;; branch if not equal
dec r17
brne waitloopl2
dec r18
```

```
brne waitloop13
pop r18
pop r17
pop r16
ret
```

what we don't is the ifdef. so we call the waiting loop only if the definition gives that.

like the typical einstellung - intention what ever, to build the city one half, than the other, we do it like this:
we don't the PAP now. and the next is: to make: rs-232, v.24.

ok, if we have this, we can have trainings for:
external interrupts, and the timer/counter is a little more excive - .... it is not hard. i now TIMSK - Timer/Interrupt Mask Register. it is for activating the TCCRA TCCRB1/TCCRB2 these are controll Register, For Counter A and B. But we have to go on.
so. the LCD - HD44780 we let out of the process now.
so: the next thing is: we make the formulars. and we have to: do the Breath First Search, as second step and the Depth First Search is no problem, form algorithmics so.

8 next

The Interpolation, Lagrange

$$l_i(x) = \prod_{\substack{j=0 \\ i \neq j}}^{n} \frac{x - x_j}{x_i - x_j} \tag{24}$$

$$p(x) = \sum_{i=0}^{n} y_i \cdot l_i(x) \tag{25}$$

$$\tag{26}$$

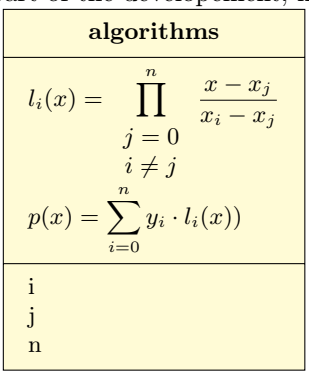
this is a very hard example to calculate and it entstand - while i try to make integration excersises with riemann and i wanted to make the square put in the diagramm optimal. and i have understood than. that is a typicall algorithm. but we can do, after, we have now this formula, and wanted to train UML to put this formula in the UML of circuitikz, tikzpicture

algorithms
LagrangeInterpolation$((y_0, y_1, \ldots, y_n), (x_0, x_1, \ldots, x_n))$
i j n

this is one thing, and the other is: to make the mathematical formular in LaTeXMathematical Expression in the thing. and what is part of the development. is: the LaTeX. Because the tikzpicture and the UMLCLASS better

\umlclass{}{}{}

is part of the developement, in this case. so



this is not part of the developement in UML, to put the algorithm in Code in the UML Diagramm, but because it is mathematical formula - we can.
this looks ok. the next thing is.
we can have a little calculation. expected value. we can easilly transform. we can transform matrix multiplication. a very simple operation in this case.
and we can put it here. what will follow. the expected value is nice, but what is in s - Standardabweichung - standard deviation, is σ a small sigma. and the formula doesn't fit always. so let's build the city in one half and another - the next thing is: we build the random calculation in more than half steps. bud buddy. now will the matrix multiplication and the: formula, expected value and the standard deviance. now!
and what is to with our assembler in amd64, i586, ... we have a std. code, for putting out a text. and counting. and not the signals! but system call usual things. and there is sth. more: we have forgotten in our atmega8 code! to! make

```
lpm
spm
```

these are very important for the std. code/text memory - which is hard. in the controller integrated - flash memory - the programm memory. and we have the typicall, commands

```
ldi
ld
ldd
lds

st
std
sts
```

for the RAM - the Working - Memory - ARBEITSSPEICHER - in german - and the: RAM. ok. and this is nice, because of a string - lot's of character, it is interessting, because, the Hard Memory - has a two byte commands
and it is a very thing: to address the hard memory with 2 byte steps. and the RAM with one byte. and the counting up in the Memory. so, these is missing at the atmega8 codes. and the next thing is:

```
;; (c) david vajda
;; atmega8/m8 codes for excersize (LPM/SPM)
;; 11/10/2026

.include "m8def.inc"

ldi r16, HIGH (RAMEND)
out SPH, r16
ldi r16, LOW (RAMEND)
out SPL, r16

ldi r16, 0xff
out DDRB, r16

main01out:
ldi YH, HIGH(2*str)
ldi YL, LOW(2*str)
main01in:
lpm r16, Y+
out PORTB, r16
cpi r16, 0x00
brne main01in
rjmp main01out

str: db "david@https://www.vajda-breadboard.de", 10, 13, 0
```

the problem of the all things is, we must not take here:

```
ldi ld ldd
```

because these are for the RAM. this is the difference. we can not put the message in the RAM, cause. the Programm and the hard thinks, are in the hard memory. and with a PC is the task/the bootloader in the BIOS must to load the programm but the Data in the RAM.
so let's go on. we can copy the Message in the Hard Memory - were it is - but - you have - take on the 2 byte limits - and countings - but the Y Ptr. just can be use to point out every one byte. but: this LPM and SPM. for the programm memory which is flash. let's go on

```
;; (c) david vajda
;; atmega8/m8 codes for excersize (LPM/SPM)
;; 11/10/2026

.include "m8def.inc"

ldi r16, HIGH (RAMEND)
out SPH, r16
ldi r16, LOW (RAMEND)
out SPL, r16

ldi r16, 0xff
out DDRB, r16
```

```
main01out:
ldi YH, HIGH(2*str)
ldi YL, LOW(2*str)
main01in:
lpm r16, Y+
out PORTB, r16
cpi r16, 0x00
brne main01in
rjmp main01out

str: db "david@https://www.vajda-breadboard.de", 10, 13, 0

;; here are very much problem with me. cause:
.data
buf: .byte ...
```

the problem is like always - also in i586, x86, nasm and the hole syntax, of segment, section, we have the difference. is there a dot before the byte, word and were we use, define byte define word and so on. and the DUP command in Intel i know, but this is a difference with atmega8 again. so let's go on. we can perform the copy bytes from the Flash Memory to the RAM. this we have to know. with out the other things.

```
;; (c) david vajda
;; atmega8/m8 codes for excersize (LPM/SPM)
;; 11/10/2026

.include "m8def.inc"

ldi r16, HIGH (RAMEND)
out SPH, r16
ldi r16, LOW (RAMEND)
out SPL, r16

ldi r16, 0xff
out DDRB, r16

main01out:
ldi YH, HIGH(2*str)
ldi YL, LOW(2*str)
ldi ZH, HIGH(buf)
ldi ZL, LOW (buf)
main01in:
lpm r16, Y+
st Z+, r16
cpi r16, 0x00
brne main01in
rjmp main01out
```

```
str: db "david@https://www.vajda-breadboard.de", 10, 13, 0

;; here are very much problem with me. cause:
.data
buf: .byte ...
```

ok, maybe there are hard specific architecture errors here. because. of the. thing is Z Pointer can not be used in every case, like u want, maybe there are some errors. ok: next thing is: now we make an intel code

we do it for linux, like always, with the INT 0x80, syscall, and the syscall is existing for Mac, and the EXIT, FORK, READ, WRITE, little differ, from the number, not really but: they have a 2 at the most significant decimal/ hexadecimal digit

```
;; (c) david vajda
;; nasm/i586/std code, for output str/msg
;; 11/10/2026

global _start
.data
        msg: db "david@https://www.vajda-breadboard.de/", 13, 10, 0
.text
_start:
        mov     rcx,    0x00
        mov     rsi,    msg
loop1:
        inc     rcx
        mov     ah,     [rsi]
        cmp     ah,     0x00
        je      loop1
;;      jz      loop1

        mov     rax,    0x04
        mov     rbx,    0x01
        mov     rdx,    rcx
        mov     rcx,    msg
        int     0x80

        mov     rax, 0x01
        mov     rbx, 0x00
        int     0x80
```

Then it should work like this. Once we have that, here's the next step: the same process applies to Mac OS X—with one difference: it uses syscalls, along with the RDI and RSI registers.

```
;; (c) david vajda
;; nasm/i586/std code, for output str/msg
;; 11/10/2026

global _start
.data
```

```
        msg: db "david@https://www.vajda-breadboard.de/", 13, 10, 0
.text
_start:
        mov     rcx,    0x00
        mov     rsi,    msg
loop1:
        inc     rcx
        mov     ah,     [rsi]
        cmp     ah,     0x00
        je      loop1
;;      jz      loop1

        mov     rax,    0x2000004
        mov     rsi,    0x01
        mov     rdi,    rcx
        mov     rcx,    msg
        syscall

        mov     rax, 0x2000001
        mov     rbx, 0x00
        syscall
```

...something like that—this is a translation using MACHO64 (that's what it's called on Mac). But now we need to return to mathematics—specifically, to matrices. In other words, matrix multiplication. As usual, this ...so we have this done. and now the next thing comes. the next thing is: matrix multiplication. and what is missing? the linear congruency we can - calculate and we can use the dice - what ever - the word is, and we can use without this the - Expected Value - and variance and standard deviation. now!

$$m = 4, n = 4, s = 4$$

$$\begin{pmatrix} 5 & 6 & 8 & 2 \\ 0 & 9 & 1 & 8 \\ 8 & 0 & 6 & 7 \\ 8 & 2 & 5 & 0 \end{pmatrix} \begin{pmatrix} 6 & 2 & 6 & 6 \\ 4 & 0 & 6 & 8 \\ 0 & 0 & 1 & 7 \\ 6 & 1 & 5 & 6 \end{pmatrix} = \begin{pmatrix} 66 & 12 & 84 & 146 \\ 84 & 8 & 95 & 127 \\ 90 & 23 & 89 & 132 \\ 56 & 16 & 65 & 99 \end{pmatrix}$$

M [1,1] = 5*6 +

we should speak in english maybe it makes more relative. cause the formular is wrong that mathematics are just without names. I say: but say sorry to your self. and forget it. it is not economy - to let the people not cry - and make them not free - because they have found a park place - with - a - let's say - sorry - a mathematicians - can speak out words - and doesn't remember - cause they irrelavant - so he can say - even - if you don't believe you can't forget. this is your problem. economy is part of mathematian. cause you can forget your problems - you can say sorry - you must not think always about that - economics don't it is good - to make the airplane - that every airplan slows down. and: if the policitians think this is serious - this maybe sounds - like a problem but it isnt'. speaking in english and learn every - numbering system and counting is

neccessary - cause, the mathematicians - use language. the number has names. so we go on.

now: we need: because we have UML. this is very important - the the methods are implemented. this is the first. it is good to make an excersize wih matrician multiplication. without any LaTeXgood looking. the UML is part of LaTeXon the one side this is: coding, when we put in:

```
\umlclass {}{}{}
```

the other thing is, the real computer coded formel in different formal languages. and another thing is: to make LaTeXmathematician Formular. This is part of programming, with LaTeX. But mathematician formular, you can put in a UML diagramm and another is the code it's self. like you see: with lagrange. and you see with expected value. and std. deviance. and linear congruency. the formula mathematics fits best in to LaTeXand UML without putting it in with code.
so let's go first. an excersize.

$$
\begin{aligned}
M\ [1\ ,1]\ &=\ 5{*}6\ +\ 6{*}4\ +\ 8{*}0\ +\ 2{*}6\ = \\
M\ [1\ ,2]\ &=\ 5{*}2\ +\ 6{*}0\ +\ 8{*}0\ +\ 2{*}1\ = \\
M\ [1\ ,3]\ &=\ 5{*}6\ +\ 6{*}6\ +
\end{aligned}
$$

what is the right expression for: 4 times 5? or 4 mal 5. it bigins to rain. sorry about my iBook. But. I Say. if we speek hindu. and now one word, and the rest is german! than we speak from our world watch? what is the right name. we now! that will not find the right words in german! there are: some things - we are usualy say - or some things we don't. mathematics are part of the german, and: english language - it's not uncool every counting. I go on another. without anxiaties. there was a strange: das bin ich nicht. no problem. we go on calculating.

$$
\begin{aligned}
M\ [1\ ,1]\ &=\ 5{*}6\ +\ 6{*}4\ +\ 8{*}0\ +\ 2{*}6\ = \\
M\ [1\ ,2]\ &=\ 5{*}2\ +\ 6{*}0\ +\ 8{*}0\ +\ 2{*}1\ = \\
M\ [1\ ,3]\ &=\ 5{*}6\ +\ 6{*}6\ +
\end{aligned}
$$

it's raining! not good. ok let's go further. the excersize we have to do.

$$
\begin{aligned}
M\ [1\ ,1]\ &=\ 5{*}6\ +\ 6{*}4\ +\ 8{*}0\ +\ 2{*}6\ = \\
M\ [1\ ,2]\ &=\ 5{*}2\ +\ 6{*}0\ +\ 8{*}0\ +\ 2{*}1\ = \\
M\ [1\ ,3]\ &=\ 5{*}6\ +\ 6{*}6\ +\ 8{*}1\ +\ 2{*}5\ = \\
M\ [1\ ,4]\ &=\ 5{*}6\ +\ 6{*}8\ +\ 8{*}7\ +\ 2{*}6\ = \\
\\
M\ [2\ ,1]\ &=\ 0{*}6\ +\ 9{*}4\ +\ 1{*}0\ +\ 8{*}6\ = \\
M\ [2\ ,2]\ &=\ 0{*}2\ +\ 9{*}0\ +\ 1{*}0\ +\ 8{*}1\ = \\
M\ [2\ ,3]\ &=\ 0{*}6\ +\ 9{*}6\ +\ 1{*}1\ +\ 8{*}5\ = \\
M\ [2\ ,4]\ &=\ 0{*}6\ +\ 9{*}8\ +\ 1{*}7\ +\ 8{*}6\ = \\
\\
M\ [3\ ,1]\ &=\ 8{*}6\ +\ 0{*}4\ +\ 6{*}0\ +\ 7{*}6\ = \\
M\ [3\ ,2]\ &=\ 8{*}2\ +\ 0{*}0\ +\ 6{*}0\ +\ 7{*}1\ = \\
M\ [3\ ,3]\ &=\ 8{*}6\ +\ 0{*}6\ +\ 6{*}1\ +\ 7{*}5\ = \\
M\ [3\ ,4]\ &=\ 8{*}6\ +\ 0{*}8\ +\ 6{*}7\ +\ 7{*}6\ = \\
\\
M\ [4\ ,1]\ &=\ 8{*}6\ +\ 2{*}4\ +\ 5{*}0\ +\ 0{*}6\ =
\end{aligned}
$$

```
M [4,2] = 8*2 + 2*0 + 5*0 + 0*1 =
M [4,3] = 8*6 + 2*6 + 5*1 + 0*5 =
M [4,4] = 8*6 + 2*8 + 5*7 + 0*6 =
```

the better algorithm is to put out all products with 0.

```
M [1,1] = 5*6 + 6*4 + 2*6 =
M [1,2] = 5*2 + 2*1 =
M [1,3] = 5*6 + 6*6 + 8*1 + 2*5 =
M [1,4] = 5*6 + 6*8 + 8*7 + 2*6 =

M [2,1] = 9*4 + 8*6 =
M [2,2] = 8*1 =
M [2,3] = 9*6 + 1*1 + 8*5 =
M [2,4] = 9*8 + 1*7 + 8*6 =

M [3,1] = 8*6 + 7*6 =
M [3,2] = 8*2 + 7*1 =
M [3,3] = 8*6 + 6*1 + 7*5 =
M [3,4] = 8*6 + 6*7 + 7*6 =

M [4,1] = 8*6 + 2*4 =
M [4,2] = 8*2 =
M [4,3] = 8*6 + 2*6 + 5*1 =
M [4,4] = 8*6 + 2*8 + 5*7 =

now we can kill - terminate - abort - not - but - reduce all with 1*x products
all the x*1

cumatative principle - what is the english expression
assioziation
distribution law


M [1,1] = 5*6 + 6*4 + 2*6 =
M [1,2] = 5*2 + 2 =
M [1,3] = 5*6 + 6*6 + 8 + 2*5 =
M [1,4] = 5*6 + 6*8 + 8*7 + 2*6 =

M [2,1] = 9*4 + 8*6 =
M [2,2] = 8 =
M [2,3] = 9*6 + 1 + 8*5 =
M [2,4] = 9*8 + 7 + 8*6 =

M [3,1] = 8*6 + 7*6 =
M [3,2] = 8*2 + 7 =
M [3,3] = 8*6 + 6 + 7*5 =
M [3,4] = 8*6 + 6*7 + 7*6 =

M [4,1] = 8*6 + 2*4 =
M [4,2] = 8*2 =
```

M [4,3] = 8*6 + 2*6 + 5 =
M [4,4] = 8*6 + 2*8 + 5*7 =

now we can make a table - wich makes it simplify

	1	2	3	4	5	6	7	8	9
1									
2								16	
3									
4									
5									
6					30			48	
7									
8									
9									

this is not a good solution, or maybe, we make a list, like in informatics, we can make pairs.

(1, 5*6), (2, 6*4), (3, 2*6), (4, 2*6),

this is not a good idea.

M [1,1] = 5*6 + 6*4 + 2*6 =
M [1,2] = 5*2 + 2 =
M [1,3] = 5*6 + 6*6 + 8 + 2*5 =
M [1,4] = 5*6 + 6*8 + 8*7 + 2*6 =

M [2,1] = 9*4 + 8*6 =
M [2,2] = 8 =
M [2,3] = 9*6 + 1 + 8*5 =
M [2,4] = 9*8 + 7 + 8*6 =

M [3,1] = 8*6 + 7*6 =
M [3,2] = 8*2 + 7 =
M [3,3] = 8*6 + 6 + 7*5 =
M [3,4] = 8*6 + 6*7 + 7*6 =

M [4,1] = 8*6 + 2*4 =
M [4,2] = 8*2 =
M [4,3] = 8*6 + 2*6 + 5 =
M [4,4] = 8*6 + 2*8 + 5*7 =

maybe - we can substitute our 5*6 by 30.

M [1,1] = 30 + 6*4 + 2*6 =
M [1,2] = 5*2 + 2 =
M [1,3] = 30 + 6*6 + 8 + 2*5 =
M [1,4] = 30 + 6*8 + 8*7 + 2*6 =

```
M [2,1] = 9*4 + 8*6 =
M [2,2] = 8 =
M [2,3] = 9*6 + 1 + 8*5 =
M [2,4] = 9*8 + 7 + 8*6 =

M [3,1] = 8*6 + 7*6 =
M [3,2] = 8*2 + 7 =
M [3,3] = 8*6 + 6 + 7*5 =
M [3,4] = 8*6 + 6*7 + 7*6 =

M [4,1] = 8*6 + 2*4 =
M [4,2] = 8*2 =
M [4,3] = 8*6 + 2*6 + 5 =
M [4,4] = 8*6 + 2*8 + 5*7 =
```

like, now we substitute the 8*6 and some other. and six times 8

```
M [1,1] = 30 + 6*4 + 2*6 =
M [1,2] = 5*2 + 2 =
M [1,3] = 30 + 6*6 + 8 + 2*5 =
M [1,4] = 30 + 48 + 8*7 + 2*6 =

M [2,1] = 9*4 + 48 =
M [2,2] = 8 =
M [2,3] = 9*6 + 1 + 8*5 =
M [2,4] = 9*8 + 7 + 48 =

M [3,1] = 48 + 7*6 =
M [3,2] = 8*2 + 7 =
M [3,3] = 48 + 6 + 7*5 =
M [3,4] = 48 + 6*7 + 7*6 =

M [4,1] = 48 + 2*4 =
M [4,2] = 8*2 =
M [4,3] = 48 + 2*6 + 5 =
M [4,4] = 48 + 2*8 + 5*7 =
```

now some other:

```
M [1,1] = 30 + 24 + 12 =
M [1,2] = 10 + 2 =
M [1,3] = 30 + 36 + 8 + 10 =
M [1,4] = 30 + 48 + 56 + 12 =

M [2,1] = 36 + 48 =
M [2,2] = 8 =
M [2,3] = 54 + 1 + 40 =
M [2,4] = 72 + 7 + 48 =
```

```
M [3,1] = 48 + 42 =
M [3,2] = 16 + 7 =
M [3,3] = 48 + 6 + 35 =
M [3,4] = 48 + 42 + 42 =

M [4,1] = 48 + 8 =
M [4,2] = 16 =
M [4,3] = 48 + 12 + 5 =
M [4,4] = 48 + 16 + 35 =

the next step, we sort, by value from great to less

similar pairs or assiociative? the second is better.
M [1,1] = (30 + 24) + 12 = 54 + 12
M [1,2] = 10 + 2 = 12
M [1,3] = (36 + 30) + (8 + 10) = 66 + 18
M [1,4] = (56 + 48) + (30 + 12) = 104 + 42

M [2,1] = 48 + 36 = 84
M [2,2] = 8 =
M [2,3] = (54 + 40) + 1 = 95
M [2,4] = (72 + 48) + 7 = 120

M [3,1] = 48 + 42 = 90
M [3,2] = 16 + 7 = 23
M [3,3] = (48 + 35) + 6 = 89
M [3,4] = 48 + (42 + 42) = 84 + 48 = 132

M [4,1] = 48 + 8 = 56
M [4,2] = 16
M [4,3] = (48 + 12) + 5 = 65
M [4,4] = (48 + 35) + 16 = 99

this is

M1 = (12, 12, 18, 42)
M2 = (84, 8, 95, 120)
M3 = (90, 23, 89, 132)
M4 = (56, 16, 65, 99)

ok, than:

M = (M1, M2, M3, M4)
M = ((12, 12, 18, 42),
        (84, 8, 95, 120),
        (90, 23, 89, 132),
        (56, 16, 65, 99))
and we compare
```

LaTeXthere are some mistakes. and we put it into latex, but we put the formular

for matrix multiplication. cause of the UML Class in the UML class in mathematics LaTeXexpressions. like the linear congruency additive congruency - like the - hash tables - like the lagrange and so on - we put it into the UML Class with LaTeXand tikzpicture. and than. we go on. and the formular for Matrix Multiplication is:

$$(m_{mn}) = \sum_{i=0}^{n} (a_{ni} \cdot b_{im}) \tag{27}$$

$$\tag{28}$$

maybe the formula is not perfect. but we put it in this style in the UML diagram.

mathematics
$(a_i) = ((a_{i-1} \cdot b + 1) \mod m)$ $h = (v_{i-1} \cdot b + h) \mod m)$ $(m_{mn}) = \sum_{i=0}^{n} (a_{ni} \cdot b_{im})$ $l_i(x) = \prod_{\substack{j=0 \\ j \neq i}} \frac{x - x_j}{x_i - x_j}$ $\sum_{i=0}^{n} l_i(x) \cdot y$ $E(x) = \sum_{i=0}^{n} p_i(X = x_i) \cdot (X = x_i)$
m n i j $\{p_0(X = x_0), p_1(X = x_1), \ldots, p_n(X = x_n)\}$ $\{x_0, x_1, \ldots, x_n\}$ $\{y_0, y_1, \ldots, y_n\}$

wich works like this the next thing is to make the depth first search, and the breath first search. real. it goes like this. first a description.

```
A: we put $r_0$ a node in the Graph in the Queue Q
B: we put the head of the queue Q in $r$
C: we take the neighbours of $r$
D: we put the first neighbours of $r$ on the queue Q ...
        ... and we do it with all
        and after that we go to C
```

and now we try to write this in python a code. maybe with this wich makes components and sub graphs we will get the problem an eulertour graph - with cycles - we make BFS - with: we make this component - and on some nodes - we have decided before we make a Valence list - and the component in the graph is the same - without the difference - that in some nodes/vertices we: make a 4/6 even difference valence, we weichen ab. so. but the code in python3 maybe.

```
## (c) david vajda
## python3 - BFS - Breath First Search
```

11/10/2026

```
V = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
E = []

Q.append (V[0])
while ((r = Q.get (V)).isnotempty ()):
        while (w = neighbours (r)):
                Q.put (w)
```

we should of adjacence matrix. we can - like I did say. but. and we go on. the difference is maybe - that: we have no

```
pred
component
```

and this could work without component - because this is the new graph, subgraph, a possibility in the Graph. And the other thing, it is not written in C, and the other Thing: it is node done with adjacence matrix. now. we make a KMP - Knuth Morris Pratt - searching in Strings - or text string - text zeichenfolgen, linear text strings

```
kmpsearch (char *a, char *p) {
        int i;
        int j;
        int N = strlen (a);
        int M = strlen (p);

        kmpinit (p);

        for (i = 0, j =-1; (i < N) && (j < M);  i++, j = next [j]) {
        // why not this. this could be a possibility without incrementing
        // j - to take j from next [j] that should be initialised in the from
        // most significant to lowest - but in the position from low to high
                while ((a [i] != p [j]) && (j > 0))
                        j = next [j];
        }
}
// and the kmpinit

kmpinit (char *p) {
        int i;
        int j;
        int M = strlen (p);

        for (i = 0;  (i < M) && (j < M); i++, j = next [j]) {
                while ((p [i] != p [j]) && (j > 0))
                        next [j] = j;
        }
}

// but there is a little confusion i cee
```

cause. now, we should turn arround the things. cause

```
kmpsearch (char *a, char *p) {
        int i;
        int j;
        int N = strlen (a);
        int M = strlen (p);

        kmpinit (p);

        for (i = 0, j =-1; (i < N) && (j < M);  i++, j++) {
        // why not this. this could be a possibility without incrementing
        // j - to take j from next [j] that should be initialised in the from
        // most significant to lowest - but in the position from low to high
                while ((a [i] != p [j]) && (j > 0))
                        j = next [j];
        }
}
// and the kmpinit

kmpinit (char *p) {
        int i;
        int j;
        int M = strlen (p);

        for (i = 0;  (i < M) && (j < M); i++, next [j] = j) {
                while ((p [i] != p [j]) && (j > 0))
                        j = next [j];
        }
}

// but there is a little confusion i cee
```

we have to think about that. this makes scence. not is logic in German. cause, the inner loop means. when: the one side - needle - and the text, and the pattern and the text, and the needle we search in the text. an in the pre run, we search the needle in the needle. maybe. than the other thing is the text. the left side - what ever.
and the difference - the inner loop: means: there is no equlancy - it is not equal. so: the j is always taken from the table - next. when there is no equality. and outer - in the prerun - the table is set to the j. cause, the j works
this is not - to: find the pattern anywhere it is - to find the pattern in the pattern but this is - bezogen auf from the beginning of the pattern - always. always find the pattern in the pattern. that means - when ever, the beginning of the pattern is found in the pattern which is is a text, to find that.
and this means - as long it is equal - the table is set to j. while it is equal - and the outer loop is the loop where it is equal because the inner is not equal.

so, now we go on. in think this form of algorithm is right

and we not make a trial again. in this situation.

by this algorithm.

and yes maybe a little quine mc cluskey. like always. and we do a bash programm. so let's start.

9 quine mc cluskey - DNF

```
(C) David Vajda
Tue Nov 10 16:22:56 2026
4 Network - TTL - Disjunktive Normalform
n   x3   x2   x1   x0   y
 0  0  0  0  0      1
 1  0  0  0  1      1
 2  0  0  1  0      1
 3  0  0  1  1      0
 4  0  1  0  0      0
 5  0  1  0  1      1
 6  0  1  1  0      0
 7  0  1  1  1      1
 8  1  0  0  0      1
 9  1  0  0  1      0
10  1  0  1  0      1
11  1  0  1  1      1
12  1  1  0  0      1
13  1  1  0  1      1
14  1  1  1  0      0
15  1  1  1  1      0
```

(C) David Vajda					
Tue Nov 10 16:22:56 2026					
4 Network - TTL - Disjunktive Normalform					
n	x_3	x_2	x_1	x_0	y
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	0

we are cleaning up the zeros:

```
(C) David Vajda
Tue Nov 10 16:22:56 2026
4 Network - TTL - Disjunktive Normalform
x3   x2   x1   x0   y
 0  0  0  0  0      1
 1  0  0  0  1      1
 2  0  0  1  0      1
 5  0  1  0  1      1
 7  0  1  1  1      1
 8  1  0  0  0      1
10  1  0  1  0      1
11  1  0  1  1      1
12  1  1  0  0      1
13  1  1  0  1      1
```

now we are forming the groups and sorting by parity (but counting not odd/even). the number of ones. now

```
(C) David Vajda
Tue Nov 10 16:22:56 2026
4 Network - TTL - Disjunktive Normalform
x3   x2   x1   x0   y
Group 0 =
 0  0  0  0  0      1
Group 1 =
 1  0  0  0  1      1
 2  0  0  1  0      1
 8  1  0  0  0      1
Group 2 =
 5  0  1  0  1      1
```

```
 10  1  0  1  0      1
 12  1  1  0  0      1
Group 3 =
 7  0  1  1  1      1
11  1  0  1  1      1
13  1  1  0  1      1
```

and we are verschmelzen diese gruppen:

```
(C) David Vajda
Tue Nov 10 16:22:56 2026
4 Network - TTL - Disjunktive Normalform
x3   x2   x1   x0   y
Group 0 =
 0  0  0  0  0      1
Group 1 =
 1  0  0  0  1      1
 2  0  0  1  0      1
 8  1  0  0  0      1
Group 2 =
 5  0  1  0  1      1
 10  1  0  1  0      1
 12  1  1  0  0      1
Group 3 =
 7  0  1  1  1      1
11  1  0  1  1      1
13  1  1  0  1      1


((0,1),(0,0,0,x))
((0,2),(0,0,x,0))
((0,8),(x,0,0,0))
((1,5),(0,x,0,1))
((2,10),(x,0,1,0))
((8,10),(1,0,x,0))
((8,12),(1,x,0,0))
((5,7),(0,1,x,1))
((5,13),(x,1,0,1))
((10,11),(1,0,1,x))
((12,13),(1,1,0,x))

(0,1)              0        0        0        x
(0,2)              0        0        x        0
(0,8)              x        0        0        0
(1,5)              0        x        0        1
(2,10)   x         0        1        0
(8,10)   1         0        x        0
(8,12)   1         x        0        0
(5,7)              0        1        x        1
(5,13)   x         1        0        1
(10,11)  1         0        1        x
```

(12,13)	1	1	0	x	

and this is not unbedingt the better schreibweise. this is to sort

(0,1)		0	0	0	x
(10,11)	1	0	1	x	
(12,13)	1	1	0	x	

(0,2)		0	0	x	0
(8,10)	1	0	x	0	
(5,7)		0	1	x	1

(1,5)		0	x	0	1
(8,12)	1	x	0	0	

(0,8)		x	0	0	0
(2,10)	x	0	1	0	
(5,13)	x	1	0	1	

ok: and now, we are bringing them together.

Group 0 =					
(0,1)		0	0	0	x
Group 2 =					
(10,11)	1	0	1	x	
(12,13)	1	1	0	x	

Group 0 =					
(0,2)		0	0	x	0
Group 1 =					
(8,10)	1	0	x	0	
Group 2 =					
(5,7)		0	1	x	1

Group 1 =					
(1,5)		0	x	0	1
(8,12)	1	x	0	0	

Group 0 =					
(0,8)		x	0	0	0
Group 1 =					
(2,10)	x	0	1	0	
Group 2 =					
(5,13)	x	1	0	1	

so in the next step, we do the following: we are bringing them together.

Group 0 =					
(0,1)		0	0	0	x
Group 2 =					
(10,11)	1	0	1	x	
(12,13)	1	1	0	x	

Group 0 =
(0,2) 0 0 x 0

Group 1 =
(8,10) 1 0 x 0

(0,2,8,10) x 0 x 0

Group 2 =
(5,7) 0 1 x 1

Group 1 =
(1,5) 0 x 0 1
(8,12) 1 x 0 0

Group 0 =
(0,8) x 0 0 0

Group 1 =
(2,10) x 0 1 0
(0,8,2,10) x 0 x 0

Group 2 =
(5,13) x 1 0 1

and now we are freeing of the rest. is there anymore or what ever.

(0,1) 0 0 0 x
(10,11) 1 0 1 x
(12,13) 1 1 0 x
(0,2,8,10) x 0 x 0
(5,7) 0 1 x 1
(1,5) 0 x 0 1
(8,12) 1 x 0 0

(0,8,2,10) x 0 x 0

(5,13) x 1 0 1

and there is a double - we have to take out

(0,1) 0 0 0 x
(10,11) 1 0 1 x
(12,13) 1 1 0 x
(0,2,8,10) x 0 x 0
(5,7) 0 1 x 1
(1,5) 0 x 0 1
(8,12) 1 x 0 0
(5,13) x 1 0 1

and the ... again. we have to bring them together, where the x is at the same position

(0,1) 0 0 0 x

(10,11)	1	0	1	x			
(12,13)	1	1	0	x			
(5,7)		0	1	x	1		
(1,5)		0	x	0	1		
(8,12)	1	x	0	0			
(5,13)	x	1	0	1			
(0,2,8,10)		x	0	x	0		

and bringing them really together.

Group 0 =							
(0,1)		0	0	0	x		
Group 2 =							
(10,11)	1	0	1	x			
(12,13)	1	1	0	x			
Group 2 =							
(5,7)		0	1	x	1		
Group 1 =							
(1,5)		0	x	0	1		
(8,12)	1	x	0	0			
Group 0 =							
(0,2,8,10)		x	0	x	0		
Group 2 =							
(5,13)	x	1	0	1			

and the next thing, goes up, what is ... we put together.

(0,1)			0	0	0	x	
(10,11)		1	0	1	x		
(12,13)		1	1	0	x		
(5,7)			0	1	x	1	
(1,5)			0	x	0	1	
(8,12)		1	x	0	0		
(0,2,8,10)			x	0	x	0	
(5,13)		x	1	0	1		

there is nothing more to do. and we can make primimplikantentafel.

	0	1	2	5	7	8	10
(0,1)	1	1					
(10,11)						1	1
(12,13)							
(5,7)				1	1		
(1,5)		1		1			
(8,12)					1		
(0,2,8,10)	1		1			1	1
(5,13)			1				

and we make a real table out of that.

	0	1	2	5	7	8	10	11	12	13
$(0,1)$	1	1								
$(10,11)$							1	1		
$(12,13)$									1	1
$(5,7)$				1	1					
$(1,5)$		1		1						
$(8,12)$						1			1	
$(0,2,8,10)$	1		1			1	1			
$(5,13)$				1						1

eine möglichkeit, wäre die spalten zusammen zu schieben, oder zu addieren. alles als wörter zu begreifen wie auch immer. wir begreifen es anders.

	0	1	2	5	7	8	10	11	12	13
$(0,1)$	1	1								
$(10,11)$							1	1		
$(12,13)$									1	1
$(5,7)$				1	1					
$(1,5)$		1		1						
$(8,12)$						1			1	
$(0,2,8,10)$	1		1			1	1			
$(5,13)$				1						1

10 Knuth Morris Pratt - and the brute searching in a string algorithm

$$\sum_{i=0}^{N \wedge (a_i = b_i)} b_i - a_i \tag{29}$$

$$\tag{30}$$

das ist am entwickeln. laut google ai gibt es ein so zu sagen, grosses identitätszeichen, ein gleichzeichen mit drei strichen, blos: der unterschied ist: dass es ein identitätszeichen ist, heisst: es geht um terme, und ausdrücke mehr oder weniger.

so: nächstes: jetzt machen wir eine innere und äussere Summe. die innere mit j und die äussere mit i. blos: das problem. die summe ist ja trotzdem eine summe. eine andere möglichkeit wäre ein UND. es gibt die grosse Tasse. Ein Cup. ist eine Vereinigungsmenge, in der Mengenlehre, entspricht dem UND bei der logik. blos: zunaechst ist das eine Vereinigung. auch, wir haben es ausprobiert mit MySQL die Vereinigungsmenge, schnittmenge verhalten sich wie AND und OR in der Logik. es geht auch De Morgan. Die Elemente sind entsprechend in der Menge enthalten oder nicht. Blos: ein Problem ist so oder so: die linearitaet muss gegeben sein. so:
noch eine Möglichkeit wäre die Polynomschreibweise. vielleicht. denn: ein Text ist ein grosses Polynom. der unterschied ist, jetzt liesse es sich ohne das Summenzeichen selber ausdrücken, die summe macht ja nur sinn, wenn es sich auch ohne das Summenzeichen ausdrücken lassen würde. und das wiederum beim

gleichheitszeichen einen sinn. wenn z.B.

$$(a_i = b_i) \wedge (a_{i-1} = b_{i-1}) \dots \tag{31}$$
$$\tag{32}$$

halt, wir nehmen statt b_i nämlich p_i

$$(a_i = p_i) \wedge (a_{i-1} = p_{i-1}) \dots \tag{33}$$
$$\tag{34}$$

allerdings: der unterschied besteht in der linearität. also es genügt nicht, jeden vergleich einzeln auf zu stellen, das gelingt erst durch die summe. eine andere möglichkeit wäre die summe zu verwenden, indem die vergleiche addiert werden. die vergleiche sind teil der Booleschen algebra. durch Archimides, der vor Boolean war. entsteht erst die Logik.

und ein Verlgiech $x_i == x_{i-1}$ z.B. ergibt bei wahrheit 1 und bei unwahrheit 0. das liesse sich addieren, also wäre die summe null oder 1. nimmt man alle elemente.

selbst. wenn: dieser ausdruck in dieser form nicht möglich wäre können wir es probieren, so weit wir kommen.

jetzt sehe ich hier. ich habe eine differenz gebildet. trotz der aussagen logik durch archimedes, die 1 oder 0 ergibt, ist natürlich laut computersysteme 1 und 2 immer noch abgeleitet die boolesche algebra mit wahr und falsch. anderes beispiel, die graphen theorie - tisch und stuhl - sind permutation zahlen usw. trotzdem, daraus kommt die schaltalgebra mit 0 und 1. trotzdem die differenz wiederum macht automatisch ein natuerliches zahlen 0 und 1. eigentlich laut ALU ist der vergleich ohne das element eines komperators zu benutzen was eigentlich unsinn ist, eine subtraktion. so. jetzt weiter.
als erstes eine innere und äussere Summe - statt schleifen. als erstes beispiel. oder zweites.

$$\sum_{i=0}^{N \wedge (a_i = p_i)} p_i - a_i \tag{35}$$
$$\tag{36}$$

erst das b ersetzt durch p und nun weiter.

$$\sum_{i=0}^{N} \sum_{j=0}^{M \wedge (a_i = p_j)} p_j - a_i \tag{37}$$
$$\tag{38}$$

wenn man wiederum wiederum das abbruchkriterium nimmt, koennte als gesamt vergleich das immer noch stehen lassen. egal. probieren wir einen polynomial vergleich. das ergebnis: Rabin Karp ähnliches etwas ist Hash-Table.

$$(a_i \cdot b^i = p_i \cdot b^i) \wedge (a_{i-1} \cdot b^{i-1} = p_{i-1} \cdot b^{i-1}) \ldots \tag{39}$$
$$\tag{40}$$

jetzt kann man für das i und das N und das j und das M einen ausdruck finden. und man kann die gesamte sache subtrahieren.

$$j \in \{0, 1, \ldots, M\} \tag{41}$$
$$i \in \{0, 1, \ldots, N\} \tag{42}$$
$$\tag{43}$$

$$(a_i \cdot b^j - p_j \cdot b^j) + (a_{i-1} \cdot b^{j-1} - p_{j-1} \cdot b^{j-1}) + \ldots (a_{i-j} \cdot b^0 - p_0 \cdot b^0) \tag{44}$$
$$\tag{45}$$

das lässt jetzt hier verschiedene möglichkeiten offen, wir haben die begriffe

- theorie
- modell
- methode

das ist jetzt kein toller begriff, aber vielleicht:

- theorie
- modell
- methode
- Schema.

und die sache noch mal richtig.

$$(a_i \cdot b^{M-1} - p_{M-1} \cdot b^{M-1}) + (a_{M-2} \cdot b^{M-2} - p_{M-2} \cdot b^{M-2}) + \ldots (a_{i-M} \cdot b^0 - p_0 \cdot b^0) \tag{46}$$
$$\tag{47}$$

was mir jetzt wiederum auffällt. sehr stark rabin karp. das ist schön. was mir auffällt ist. dass der eigentliche sinn in dem Suchen in Zeichenfolgen, Knuth Morris Pratt oder Grob oder Boyer Moore - geht ja von der anderen seite her. das problem: die linearität ist nicht gegeben. wir haben das an einer anderen

stelle. Fibonacci folgen z.B eine typische folge ist

$$
\begin{aligned}
a_0 &= 1 && (48)\\
a_1 &= 1 && (49)\\
a_2 &= a_1 + a_0 && (50)\\
a_3 &= a_2 + a_1 && (51)\\
a_4 &= a_3 + a_2 && (52)\\
a_5 &= a_4 + a_3 && (53)\\
\dots & && (54)\\
a_n &= a_{n-1} + a_{n-2} && (55)\\
a_{n+1} &= a_n + a_{n-1} && (56)\\
& && (57)
\end{aligned}
$$

eine andere folge.

$$
\begin{aligned}
a_0 &= 1 && (58)\\
a_1 &= 1 && (59)\\
a_2 &= a_1 + a_0 \cdot (-1)^0 && (60)\\
a_3 &= a_2 + a_1 \cdot (-1)^1 && (61)\\
a_4 &= a_3 + a_2 \cdot (-1)^2 && (62)\\
a_5 &= a_4 + a_3 \cdot (-1)^3 && (63)\\
\dots & && (64)\\
a_n &= a_{n-1} + a_{n-2} \cdot (-1)^{n-2} && (65)\\
a_{n+1} &= a_n + a_{n-1} \cdot (-1)^{n-1} && (66)\\
& && (67)
\end{aligned}
$$

eine folge. oder man die nimmt die erste. nur beispiele, aber: eine Folge ist generell mit einer Tabelle dar zu stellen, eine Abbildung auch. wir sehen, das die Hashtabelle eigentlich eine Folge und nicht nur einfach eine Abbildung ist, eine Abbildung ist eine Funktion, allerdings nicht auf die Relleen Zahlen hin sondern auf die natürlichen. Methode, ist natürlich immer die frage was das ist. ein verfahren.
Möglicherweise fasst der Begriff Methode, die Begriffe, Abbildung, Funktion und Folge zusammen. unter anderem: ist die Lineare Kongruenz, eine Methode. während die hashtabelle eigentlich das mehr oder weniger gleiche ist, von wegen: schema. nur dass statt der Summe mit 1, quasi der h oder v wert addiert wird. ebenso: sind solche zusammenhänge über tabellen aus zu drücken wobei abbildungen letzten endes immer tabellen sind. trotzdem, wie auch bei: Depth First Search und Bread First Search, sind das alles Tabellen, in folgen. nächstes. nun über diese tabelle, lässt sich jetzt folgendes tun:
die quasi lineare abarbeitung des KMP Knuth Morris Pratt, suchen in Zeichenfolgen erledigen.

ich würde allerdings vorschlagen, dass hier gewisse ausdrücke nicht gefunden sind. umgesetzt wurden und das muss passieren und zwar formal. das ist der

hauptfehler in allen denkweisen, die unterlassung des formal richtigen. schöne denkweisen - nicht nur schemen, schöne. am besten noch, wenn methode und theorie noch vorhanden ist, ganz ohne, ist das formale wiederum schlecht, aber das formale ist nie da

$$\sum_{j=0}^{M} a_{i-M+j} \cdot b^{M-j} - p_{M-j} \cdot b^{M-j} \tag{68}$$

$$\tag{69}$$

Das erinnert natürlich stark an Rabin Karp. blos: die alternative ist jetzt eine tabelle zu nehmen. wie bei jeder folge. ohne die summe selber.

so, jetzt machen wir einfach mal eine folge, die so geht. dass: wir einfach eine tabelle nehmen. und dabei: werden wir einfach mal examplerisch und machen das von oben trotzdem ende noch umgesetzt

i	j	a_i	p_j

wir sehen so geht es nicht. wir brauchen eine mathematische folge. die wird sich genauso ausdrücken, blos: eine fibonnacci folge, lässt sich als tabelle ausdrücken, liefert aber den ausdruck für die folge, mit dem unterschied. bei der summe - also der Series. mit summenzeichen. ebenso bei folge, wird trotzdem die beziehung herstellung ,nicht einfach wie bei einer abbildung, ohne zusammenhang. egal: jede abbildung auf ein Element, ohne Rekursiven, keine ernsthafte rekursion unter umständen, wie auch immer.
dann machen wir ein Element f_i

i	j	a_i	p_j	f_i
$i_k + M - 1$	$M - 1$	a_{i_k+M-1}	p_{M-1}	$a_{i_k+M-1} - p_{M-1}$

weiter komme ich jetzt hier ehrlich gesagt nicht und ich muss das problem einstellen. das problem ist an wissenschaftlichen immer weiter zu reiten und nicht zu merken, wann schluss ist. also zurück zum anfangspunkt. ich schreibe den gegebenen algorithmus auf und mehr ist nicht drin. was haben wir eigentlich?

- Theorie
- Modell
- Methode

das ist teil von algorithmischer übung, ohne erklärung
annahme:

1. Modell
2. Theorie: braucht ein Modell als grundlage
3. Methode

nächstes:

1. Modell: analoge abbildung der realiät
2. Theorie: spekulative annahme

3. Methode: vielleicht: funktion, abbildung, folge
4. Maschine
5. System
6. Mechanismus, bzw. die methode
7. und die vordere Seite - das Frontend
8. der Hintergrund - das Backend

das ist jetzt verzettelt, trotzdem möglich:

1. Modell: analoge abbildung der realiät
2. Theorie: spekulative annahme
3. Methode: vielleicht: funktion, abbildung, folge
4. Maschine
5. System
6. Mechanismus, bzw. die methode
7. und die vordere Seite - das Frontend
8. der Hintergrund - das Backend
9. der Compiler: Parsing - Formale Sprache - BNF
10. die Zielsprache
11. der mechanismus ist das Top-Down- Rekursiver Abstieg Parsing
12. wiederum: das Pattern Matching: benutzt als Frontend: den Compiler
13. der machanismus ist im frontend: Der Compier und die formale Sprache
14. auf der anderen Seite, der Automat.

das ist jetzt so ein zeug. dabei haben wir

1. die methode
2. der test!
3. aber der Test, die Simulation
4. Test und Ausführung usw.

aber jetzt sammlung:

1. Modell
2. Theorie
3. Methode

4. Test
5. Maschine
6. System
7. Automat
8. Schaltwerk
9. Darstellung
10. Simulation
11. Test

ok, ein realer Algorithmus, erst eine formale Sprache - BNF - ohne beseitigung der linksrekursion

```
<expr> ::= <term> | <expr> '+' <term>
<term> :: <factor> | <term> '*' <factor>
<factor> ::= '('<expr>')' | <id> | <const>
```

weiter: jetzt: kommt noch der KMP - Knuth Morris Pratt

```
kmpinit (char *p) {
        int i;
        int j;
        int M = strlen (p);

        for (i = 0, j = 0;  (i < M);  i++, j++, next [j] = j)
                while ((p [i] != p [j]) && (j > 0))
                        j = next [j];
        /*..*/
}
/* ... */
kmpsearch (char *a, char *p) {
        int i;
        int ;
        int N = strlen (a);
        int M = strlen (p);

        kmpinit (p);
        /* oder initnext */

        for (i = 0, j = -1; (i < N) && (j < M); i++, j++) {
                while ((a [i] != p [j]) && (j > 0))
                        j = next [j];
        }
        /* ... */
}
```

wenn man sich das spezifisch hier anguckt. ist das problem. man muss darauf achten:

1. die innere schleife, ist die, wo es nicht gleich ist.
2. und darüber hinaus, sage ich mal: wenn es nicht gleich ist, dann muss wie in dem text, eben: sagen wir mal: der index von dem muster neu gesetzt werden
3. die äussere schleife ist, ist das, wo es gleich ist, das verwechselt man hier gerne: und: hier wird next in dem vorlauf initialisiert. und das wenn es gleich ist.
4. blos das ungleiche ist in der inneren schleife. und: das heisst: die äussere - schleife. ist das wo das gleich.
5. und da wird next initialisiert und zwar mit j. deswegen weil, der unterschied, wenn es gleich ist, fangen wir von vorne. die next tabelle, initialisiert nicht irgendwie ... ...das muster was wir suchen, im text, aber im muster, wird von vorne gesucht. das ist der unterschied. also wird die next tabelle in folge. die next vorne intiilaisiert
6. das heisst: next entspricht dem j. das next wird mit initisiliert. nicht falsch verstehen, in der äusseren schleife ist es gleich. solange es gleich ist ist es j. und das geht von dem anfang aus.

11 so weit

jetzt kommen die anderen algorithmen

- selection sort
- insertion sort
- bubble sort
- shell sort (geht nicht bei mir)
- quick sort

11.1 selection sort

```
selectionsort (char s []) {
        int N = strlen (s);
        int i, j;
        char tmp;
        char min;

        for (i = 0;  i < N;  i++) {
                min = i;
                for (j = i+1;  j < N;  j++)
                        if (s [j] < s [min])
                                min = j;
                tmp = s [min];
                s [min] = s [i];
```

```
                s [i] = tmp;
        }
}
```

11.2 insertion sort

```
insertionsort (char s []) {
        int N = strlen (s);
        int i, j;
        char tmp;

        for (i = 0;  i < N;  i++) {
                for (j = i+1;  j < N;  j++) {
                        if (s [j] < s [i]) {
                                tmp = s [min];
                                s [min] = s [i];
                                s [i] = tmp;
                        }
                }
        }
}
```

bubble sort, meiner meinung nach könnte man sich zunächst vorstellen, wir tauschen element für element und machen das solange bis es nichts mehr zu tun gibt.

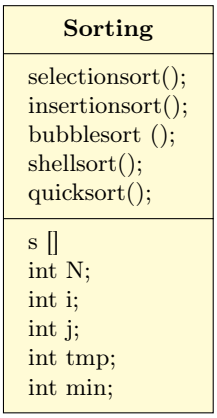
11.3 bubble sort

```
bubblesort (char s []) {
        int N = strlen (s);
        int i, j;
        int tmp;
        j = 1;

        while (j) {
                j = 0;
                for (i = 0;  i < N-1;  i++) {
                        if (s [i] > s [i+1]) {
                                tmp = s [i];
                                s [i] = s [i+1];
                                s [i+1] = tmp;
                                j = 1;
                        }
                }
        }
}
```

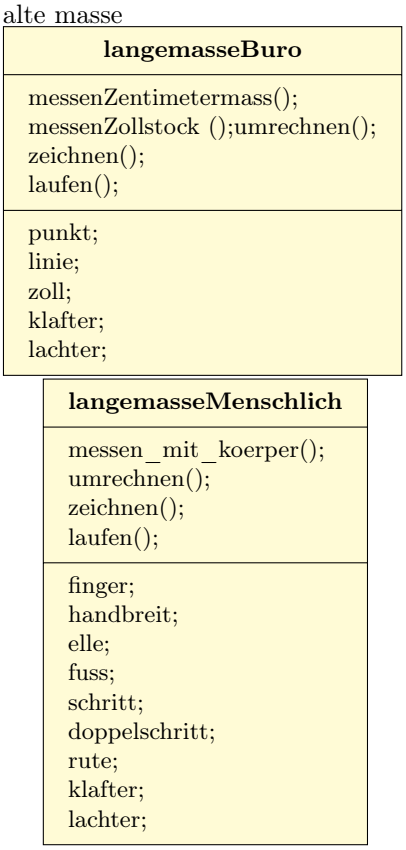
dann, der quick sort

12 UML-Klassenmodell



13 die übungsklasse

alte masse



Gewichtsmasse

wiegen();erhoehen();
erniedrigen();
teilen ();
multiplizieren();
addieren();
vergleichen ();
subtrahieren();

cent; gran; karat; skrupel; heller; pfennig; quentchen; lot; unze; mark; pfund; zentner; gewicht1; gewicht2; gewicht3;

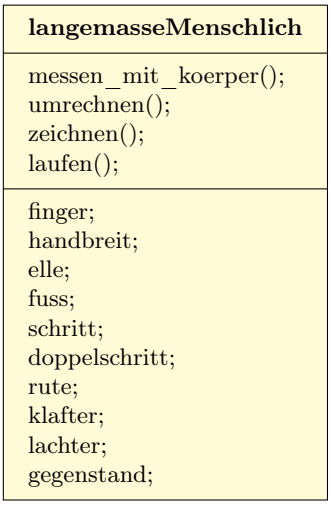
also, noch mal:

1. das erhöhen ist eigentlich inkrement
2. das erniedrigen ein dekrement
3. wenn wir: in dem punkt, linie, zoll inkrementieren, landen irgendwann in der linie, obwohl bisher bei punkt waren
4. und normalerweise haben wir eine längeneinheit und: bei gewichten, addieren wir zwei: oder vergleichen sie.

langemasseBuro

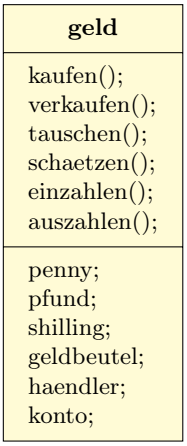
messenZentimetermass();
messenZollstock ();umrechnen();
zeichnen();
laufen();

punkt; linie; zoll; klafter; lachter; laenge1; laenge2; laenge3; cmmass;



dann können wir währungen einführen

1. wert
2. realwert
3. tauschwert
4. gebrauchswert
5. wechselkurs



das muss ich noch genauer lernen. in komponenten diagrammen würde ich sagen: würde man versuchen die klassen miteinander in verbindung zu bringen. hier ein paar chemische zusammenhänge, etwa wie viel kostet das Gran Gold, oder gleich chemische formeln. etwa: ausgehend von der Dichte/molare Masse/molares Volumen/Stoffmengenkonzentration liessen sich parameter wie druck angeben. und so weiter. und dann könnte man innere energie angeben. in dem komponenten diagramm.
frage ist wie üben wir das: vollstände kunstwerke in der software technik. oder jeden ein tag ein bisschen längenmass/holmass. . . usw. und dann kann man natürlich dann auch noch verkaufen und kaufen und einheiten umrechnen, wenn man das hat. und nicht zu letzt. von der karte ins gelände auch noch ein bisschen

geodäsie usw.

jetzt weiter üben.

14 ein wenig netzwerktechnik + signals + bash + klasse

/bin/
mv; cp chmod; chown; ;
;/dev/ttyS0; file1; file2; uid; gid;

SIGHUP
SIGINT
SIGQUIT
SIGKILL
SIGTERM
SIGILL
SIGABRT

HANGUP am Telefon
INTERRUPT am Telefon, kein Interrupt durch Devices/bzw. ein Unterbrechung am Te
QUIT: ist beendet
KILL: besonders toeten
TERMINATE: ist zeitlich und geplant
ABORT: Abnormale Beendigung

STRG–C –
STRG–D – Eingabe Beenden/Unterbrechung

Codes so im Allgemeinen
ACK – Acknowlegement
NAK – No Acknowlegement – keine Empfangsquittierung!
DLE – Data Link Escape
ENQ – Enquiry – Bestaetigung – aber nicht empfang

hier: quittung: das auf das Signal –
das andere eher: eine quittierung – das eine eher funkververkehrt
das andere eher: das objekt selber

```
ENQ
STX Start of Text
ETX End of Text
EOT End of Transmission
SOT/SOH Start of Transmission/Start of Heading
ETB End of Transmission Block
CAN Cancel
DEL Delete
ESC - Escape - das heisst, raus gehen - aus der Sache
                Cancel ist zuruecknehmen, abkanceln -
        dagegen ist ABORT: Abnormal Beenden
LF - Line Feed
FF - Form Feed
HT - Horizontal Tab
VT - Vertical Tab
CR-LF Carriage Return Line Feed
CR - Carriage Return
SI - Shift In
SO - Shift Out

Daneben Leitungen: v.24/rs232
TxD - Transmit Data
RxD - Recieve Data
RXC - Atmega8 - Recieve Complete
TXC - Atmega8 - Transmit Complete
RXEN - Atmega8 - Recieve Enable
TXEN - Atmega8 - Transmit Enable
Baudrate: Zeichen pro Sekunde (aber auf Zeichensatz bezogen)
        2400, 4800, 9600...
5 .. 9 Datenbit
1 oder 2 stopbit
1 startbit
odd/even/no parity, so weit ich weiss: ist even gerade
Terminal: datenendgeraet: computer (flughafen, ist das grosse ding)
set: modem - data set ready - am flughafen gate - hinter dem terminal

Data Set Ready
Data Terminal Ready
Clear To Send
Ready To Recieve?

Signal Ground
Frame Ground... usw.

und wichtige begriffe:
Clear
Ready
Request
Carrier - traeger
Enquriy
```

```
Acknowlegement
Custom Cleared - Zollabfertigung
ansonsten: Custom Cleared Completed
```

usw.

so. jetzt weiter. danach ein wenig netzwerkkarte. vorher allerdings: ein atmega8 programm mit PAP.

```
;; (c) david vajda
;; 11/13/2026
;; led/out/count/up/waiting loop

.include "m8def.inc"

.equ innerloop1 =       0x04
.equ outerloop1 =       0xff
.equ outerloop2 =       0xff

ldi r16, HIGH (RAMEND)
out SPH, r16
ldi r16, LOW (RAMEND)
out SPL, r16

ldi r16, 0xff
out DDRB, r16

ldi r16, 0xff
main01:
out PORTB, r16
rcall waitingloop
dec r16
rjmp main01

waitingloop:
push r16
push r17
push r18
ldi r16, 0xff
waitingloop1:
ldi r17, 0xff
waitingloop2:
ldi r18, 0x04
waitingloop3:
dec r18
brne waitingloop3
dec r17
brne waitingloop2
dec r16
brne waitingloop1
```

```
pop r18
pop r17
pop r16
ret
```

15 ein wenig klavier spiel mit metronom

der PAP kommt gleich. ASM-Diagramme + RTL-Notation sind für hardwarealgorithmen und automaten. also: das kommt für das assembler programm gleich. inzwischen klavier. unser Knuth Morris Pratt + unser nicht deterministischer automat zum zeichensuchen könnten dann noch ein wenig eine melodie wiederfinden in Noten C-D-E-F-G-A-H-C.

16 die nächste übung

was zu tun ist, ist mir aufgefallen.

- ein Graph von Tübingen nach Stuttgart - Eisenbahn - mit: dem gedachten, im Graph, von Reutlingen nach Stuttgart.
- ein periodensystem so weit es sitzt aus dem kopf
- das PAP zu vorher - zum atmega8 programm
- jeden tag circuitikz mit widerstand im kopf - bei 5V Betriebsspannung U_B. ein Widerstand mit $r\Omega$ mit Versorgungsspannung - und stromkreis in circuitikz einzeichnen
- dasselbe mit einem Transistor
- die annahme, jeden tag: bei einem Verstärkungsfaktor von 500. muss am Eingang für die Schaltung, bei Basisschaltung. z.B. und hier ist baustelle, wie ist der Verstärkungsfaktor BC548/BC549 an der Basis. Im Arbeitsstromkreis.... und trotzdem: Meinetwegen Arbeitsstromkreis: 5V U_B und: LED, meinetwegen, und Widerstand. Zum Widerstand E12. Z.B. + die farben, dann dazu. in der realität. und an der Basis, entsprechend, weil einfach: schalter. auch 5V von U_B dann entsprechend bei Verstärkungsfaktor: 500 halt den Widerstand
- daneben die anderen schaltungen: Basisschaltung, Emitterschaltung, Collectorschaltung. dann entsprechend: der Widerstand, am Emitter am Collector, an der Basis von U_B und vom Emitter oder Spannungsteiler.
- aber: immer nur so weit wie geht, sonst baustelle + Verschätzung: Nachher: MAX 737. Schon ist das ist. Baustelle.
- Hier Kurve Horizont, aber als Kurve Einschätzung, und gleich beispiel, anderes: zunächst vorraussetzungs, erwartungswert, standardabweichung σ und Varianz. und dann die Kurve. Und: heisst das von TUE nach STG. heisst das NO - Nürtingen/Oberboihingen zu erst. Oder MB - Metzingen Bempflingen. vermutlich das zweitere. In das Graphendiagramm - dann

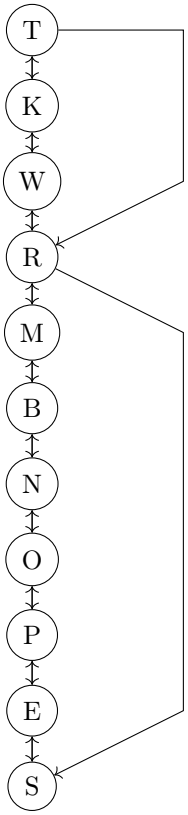
zwei Vierecke. Am Besten oben Pfeil rein und vierecke gestrichelt. Hier besteht die Möglichkeit falscheinschätzung. wie der höhenstabilisator und Warner bei MAX 737. Hier natürlich auch, weil ich nicht was das macht. hier die eine kurve, hier die andere, die gehen irgendwo auseinander - Baustelle. und da das Viereck rein. gleich anderes beispiel

- die 3 Dinge in der Wahrscheinlichkeitsrechnung: Erwartungswert, Standardabweichung, variannz + aber würfeln σ nicht vergessen, das ist ein kleines Sigma.
- dann: haben wir:
- jetzt ein Beispiel: jeden Tag in LaTeXein Timing Diagramm. Es gibt in UML: Strukturdiagramme (Klassen, Komponenten, Komposition), Verhalten (dazu gehört das Zustandsdiagramm (bei Automaten - in der Graphen Theorie - gezeichnete Graphen mit Knoten und Kanten)). und aktivitätsdiagramme mit wie auch immer - tikz. und zwar tikz-timing oder so. und das ist aber Teil von LaTeXÜbung
- an der stelle die frage: warum kann man eigentlich die sachen beherrschen, aber das nicht in LaTeXhinschreiben? also: das aber nicht ein mal! jeden Tag ein Mal! Zum Beispiel Dampfmachine, nachher, aber Haydn und so weiter. Muss ich jeden Tag Ravell übung, oder Katze liegt im Schnee, ABC
- und muss ich nicht einfach circuitikz mit üben. dabei weiss ich - IEEE-1984/91, natürlich nicht wie viele - spitzen hat ein Widerstand im IEEE-1984/91, also das auf papier üben. Eine schaltung so auf papier lohnt sich nicht. sondern mit circuitikz, und dazu gehört: tikz-Timing- so weit ich weiss, wie das heisst, dazu gehört
- der Name des Package - die Umgebung - die Syntax (ich glaube X - zum beispiel, geschweifte Klammer, H und L irgendwie und bei 8 H auf ein Mal, laengeres Taktintervall 8 H aber in Geschweifte Klammer)
- das kann man zu dem atmega8 Programm machen von gerade. und: dann in das quine Mac Cluskey + dem GAL16V8 und dann eine 5 Y Linie
- dann: das andere ist: in der chemie z.B. ein Experiment: G per M. ist wie günstig per möglich vom Namen her. wenn man das hat, weiss man gramm pro mol. und jetzt könnte, aber nur so weit man kommt. wie heisst das glas. wie ist die untere linie - für den untersten füllwert und für den oberen. dazu schreiben. dann schreiben. das gewicht und das Volumen und dann so zu sagen: Molare Masse, bei Wasser und: Dichte und Stroffmengenkonzentration
- Dann Dampfmaschine: Diese hat Innere Energie + Wärmeenergie. und die Entropie und Enthalpie. Die Entropie ist das Chaos. mehr oder weniger. In Prozent. Und: dann hat man den Druck p und jetzt ist hier einfach, MAX 737. dann Bewegungsenergie schreibe ich dazu. also: ich weiss nicht, wie sich die errechnet, eine Formelsammlung habe ich, weiss aber nicht. Ich weiss Elektrische Arbeit ist: $W = U \cdot I \cdot t$. also ist allerdings die Reaktionsenthalpie ist der Treibstroff. dieser verbrennt aber anders. weil nicht

plötzlich. die Reaktion ist Treibstoff + Sauerstoff. und geht in die Dampfmaschine und wandelt sich um über Bewegungsenergie/Rotationsenergie in Elektrische

- anderes Beispiel: ich habe ein Oszilloskop: und das hat nicht VT und HT - sondern VDIV und HDIV. Das sind die einzelnen nicht quadranten sondern MAX737. und dann geht das so: 500, 200, 100, 50, 20, 10. so ungefähr. wenn ich das schon mit circuitikz und dem widerstand und transistor übe. dann: kann ich das rein machen. Plus, dass ich die Spannung habe. bzw. Strom bzw. Leistung. und dann die Zeit rein.
- wenn die Dampfmaschine im Griff hätte, da ist MAX 737. dann weiss ich: das zeug verbrennt. Hier ist bei der Enthalpie problem. weil nicht plötzlich, die Dampfmaschine enthält der Kessel immer weniger wasser, also: wird der Druck geringer das Timing hingezogen würde bedeuten: die Enthalpie sichtbar, aber ich kann ja einzeichnen so weit ich kann, und wo fehler, fehler. so einfach weiter, ich übe
- Noch ein Bash Programm
- und noch ein Struktogramm dazu.
- ich muss Tiefensuche/Breitensuche: Depth First Search und Bread First Search üben.

16.1 Diagramm: Strecke Tübingen - Stuttgart



hier das tikzpicture:

```
\begin{tikzpicture}
\node [circle,draw=black] (T) at (0,10) {T};
\node [circle,draw=black] (K) at (0,9) {K};
\node [circle,draw=black] (W) at (0,8) {W};
\node [circle,draw=black] (R) at (0,7) {R};
\node [circle,draw=black] (M) at (0,6) {M};
\node [circle,draw=black] (B) at (0,5) {B};
\node [circle,draw=black] (N) at (0,4) {N};
\node [circle,draw=black] (O) at (0,3) {O};
\node [circle,draw=black] (P) at (0,2) {P};
\node [circle,draw=black] (E) at (0,1) {E};
\node [circle,draw=black] (S) at (0,0) {S};

\draw [->] (S) -- (E);
\draw [->] (E) -- (S);

\draw [->] (E) -- (P);
\draw [->] (P) -- (E);

\draw [->] (P) -- (O);
\draw [->] (O) -- (P);

\draw [->] (O) -- (N);
\draw [->] (N) -- (O);

\draw [->] (N) -- (B);
\draw [->] (B) -- (N);

\draw [->] (B) -- (M);
\draw [->] (M) -- (B);

\draw [->] (M) -- (R);
\draw [->] (R) -- (M);

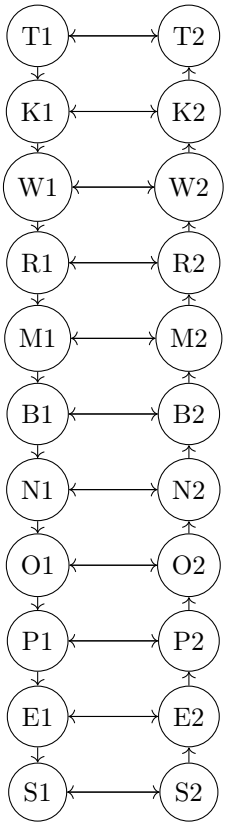
\draw [->] (R) -- (W);
\draw [->] (W) -- (R);

\draw [->] (W) -- (K);
\draw [->] (K) -- (W);

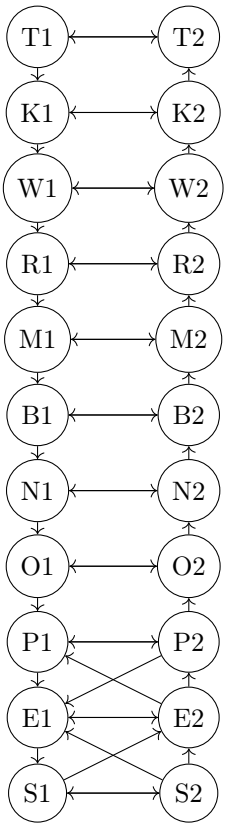
\draw [->] (K) -- (T);
\draw [->] (T) -- (K);

%\draw [->] (T) -- [edge,bend=left,angle=270] (R);

\draw [->] (T) -- (2,10) -- (2,8) -- (R);
\draw [->] (R) -- (2,6) -- (2,1) -- (S);
\end{tikzpicture}
```

jetzt noch mal, hinfahrt und rückfahrt. mit doppelgleis: 

angenommen es gäbe ein nebengleis/??? dann könnte man machen.



könnte man wahrscheinlich nie machen!!!

```
\begin{tikzpicture}
\node [circle,draw=black] (Th) at (0,10) {T1};
\node [circle,draw=black] (Kh) at (0,9) {K1};
\node [circle,draw=black] (Wh) at (0,8) {W1};
\node [circle,draw=black] (Rh) at (0,7) {R1};
\node [circle,draw=black] (Mh) at (0,6) {M1};
\node [circle,draw=black] (Bh) at (0,5) {B1};
\node [circle,draw=black] (Nh) at (0,4) {N1};
\node [circle,draw=black] (Oh) at (0,3) {O1};
\node [circle,draw=black] (Ph) at (0,2) {P1};
\node [circle,draw=black] (Eh) at (0,1) {E1};
\node [circle,draw=black] (Sh) at (0,0) {S1};

\node [circle,draw=black] (Tz) at (2,10) {T2};
\node [circle,draw=black] (Kz) at (2,9) {K2};
\node [circle,draw=black] (Wz) at (2,8) {W2};
\node [circle,draw=black] (Rz) at (2,7) {R2};
\node [circle,draw=black] (Mz) at (2,6) {M2};
\node [circle,draw=black] (Bz) at (2,5) {B2};
\node [circle,draw=black] (Nz) at (2,4) {N2};
\node [circle,draw=black] (Oz) at (2,3) {O2};
\node [circle,draw=black] (Pz) at (2,2) {P2};
\node [circle,draw=black] (Ez) at (2,1) {E2};
```

```
\node [circle,draw=black] (Sz) at (2,0) {S2};

\draw [->] (Sh) -- (Sz);
\draw [->] (Sz) -- (Sh);
\draw [->] (Sh) -- (Ez);
\draw [->] (Sz) -- (Eh);

\draw [->] (Eh) -- (Ez);
\draw [->] (Ez) -- (Eh);
\draw [->] (Ez) -- (Ph);
\draw [->] (Pz) -- (Eh);

\draw [->] (Ph) -- (Pz);
\draw [->] (Pz) -- (Ph);

\draw [->] (Oh) -- (Oz);
\draw [->] (Oz) -- (Oh);

\draw [->] (Nh) -- (Nz);
\draw [->] (Nz) -- (Nh);

\draw [->] (Bh) -- (Bz);
\draw [->] (Bz) -- (Bh);

\draw [->] (Mh) -- (Mz);
\draw [->] (Mz) -- (Mh);

\draw [->] (Rh) -- (Rz);
\draw [->] (Rz) -- (Rh);

\draw [->] (Wh) -- (Wz);
\draw [->] (Wz) -- (Wh);

\draw [->] (Kh) -- (Kz);
\draw [->] (Kz) -- (Kh);

\draw [->] (Th) -- (Tz);
\draw [->] (Tz) -- (Th);


\draw [->] (Sz) -- (Ez);
\draw [->] (Eh) -- (Sh);

\draw [->] (Ez) -- (Pz);
\draw [->] (Ph) -- (Eh);

\draw [->] (Pz) -- (Oz);
\draw [->] (Oh) -- (Ph);

\draw [->] (Oz) -- (Nz);
```

```
\draw [->] (Nh) -- (Oh);

\draw [->] (Nz) -- (Bz);
\draw [->] (Bh) -- (Nh);

\draw [->] (Bz) -- (Mz);
\draw [->] (Mh) -- (Bh);

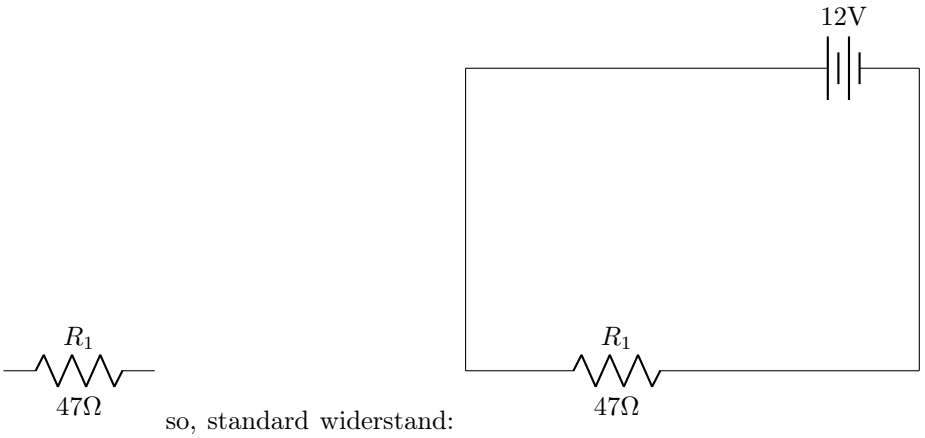
\draw [->] (Mz) -- (Rz);
\draw [->] (Rh) -- (Mh);

\draw [->] (Rz) -- (Wz);
\draw [->] (Wh) -- (Rh);

\draw [->] (Wz) -- (Kz);
\draw [->] (Kh) -- (Wh);

\draw [->] (Kz) -- (Tz);
\draw [->] (Th) -- (Kh);
\end{tikzpicture}
```

17 eine schaltung/circuit mit Widerstand/resistor



so, standard widerstand:
das muss ich jetzt üben und vorher was zu essen kaufen.

$$I \quad U \quad R \tag{70}$$

(71)

das ist die Reihenfolge U steht immer oben.

$$I = \frac{U}{R} \tag{72}$$

$$U = I \cdot R \tag{73}$$

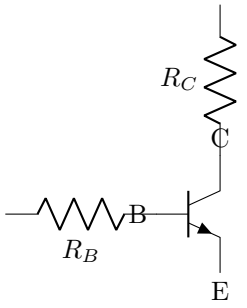
$$R = \frac{U}{I} \tag{74}$$

$$I = \frac{U}{R}, U = I \cdot R, R = \frac{U}{I}$$

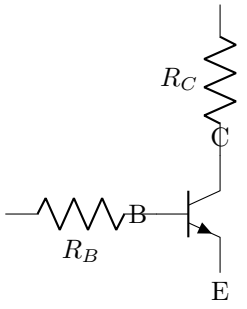
$$U_B = 12V \tag{75}$$
$$R = 47\Omega \tag{76}$$

$$U_B = 12V \tag{77}$$
$$R = 47\Omega \tag{78}$$
$$R = \frac{12V}{47\Omega} \tag{79}$$
$$R = \frac{12V}{48\Omega} \tag{80}$$
$$R = 0.25A \tag{81}$$

ok, dann haben wir das ausgerechnet.
jetzt probieren wir es mit einem Transistor.




```
\begin{circuitikz}
    \draw (0,0) node[npn] (T1) {};
    \draw (T1.base) node[anchor=east] {B}
          (T1.collector) node[anchor=south] {C}
          (T1.emitter) node[anchor=north] {E};
    \draw (T1.collector) to[R, l=$R_C$] ++(0,2);
    \draw (T1.base) to[R, l=$R_B$] ++(-2,0);
\end{circuitikz}
```



BC549.

gut jetzt farbwiderstands vergessen.

Farbe	Color	1st	2nc	$multipkier = 10^{3rd}$	tolerance
schwarz	black	0	0	1	
braun	brown	1	1	10	
rot	red	2	2	100	
orange	orange	3	3	1000	
gelb	yellow	4	4	10k	
grün	green	5	5	100k	
blau	blue	6	6	1M	
violette	purple	7	7	10M	
grau	gray	8	8	100M	
weiss	white	9	9	1000M	

nächstes: wenn wir jetzt: die tolleranz nicht angeben können. ich habe falsch anfangen und kann mir solche ersparen. sondern: also: wenn was nicht klar ist, lassen wir das offen, das kommt später, die tabelle kam nicht ein mal.

nächstes:

so, jetzt machen wir eine schaltung. wenn wir davon ausgehen, dass $U_B = 5V$ ist und der verstärkungsfaktor. 500. dann müssen wir. $\frac{5V}{500}$ nehmen, und was ist mit der Stromstärke. die erraten wir jetzt. und zwar wie? das ist ganz einfach: unsere Diode hat. also, mir fehlt die Stromstärke. und wir machen das so schnell wie möglich.

also: wenn der Widerstand bei dem Steuerstromkreis: 330Ω sind. dann müssen wir die tabelle nehmen. das haben wir vorher vermieden.

so: 47Ω brauchen wir erst mal.

das ist in Farben:

$$4,7 \cdot 10^0$$

```
gelb violette 10^0
entspricht
gelb, violette, schwarz.
```

und dann geht es weiter.

$$330\Omega \sim orange, orange 10^1 =$$

```
orange orange brown
```

ok. vermutlich. und jetzt geht es weiter.eine Diode hat:

- Sperrspannung - Grenspannung - Durchlassspannung
- obere Durchlassspannung
- Durchbruchspannung

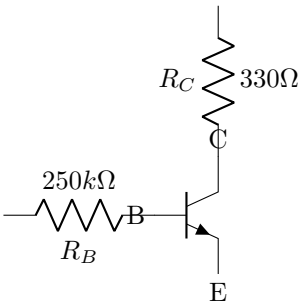
Eine Diode wird normalerweise mit 330Ω bei $5V$ betrieben. ich glaube sie hat $20mA$ aber wir nehmen es anders. weil: bei $330\Omega + \ldots$

also ein gleichungssystem. linear. Dabei haben wir:

- $U_B = 5V$
- $1.7V + \cdots = 5V$
- 330Ω
- $U = I \cdot R$
- $U_B = U_1 + U_2$
- $I = \frac{U}{R} \Rightarrow = R = \frac{U}{I}$
- $I_1 = \frac{U_1}{R_1} \Rightarrow R_1 = \frac{U_1}{I_1}$
- dabei $U_B = 5V$, $U_1 = 1.7V$. dann $I_1 = I_2$ ist das gesamte I. damit können wir die sache ausrechnen. wir haben ein I und: $5V$. so. dann haben $1.7V$
- $5V = 1.7V + 330\Omega \cdot I, 1.7V = R \cdot I$, aber anders:
- $5V - 1.7V = 330\Omega \cdot I$
- $\frac{3.3V}{330\Omega} = I$
- mit oder ohne Rechenfehler, die Rechnung passiert jeden Tag!
- und das ist I. und das I entspricht dem, was wir an der Basis für den Schalter nehmen
- so, jetzt: geht das so weiter: jetzt haben wir I ausgerechnet, das nehmen wir für die Basis
- $\frac{3.3V}{330\Omega} = 0.01A$, das passt ganz gut zu den $20mA, denn 100mA = 0.1A$, ok.
- jetzt nehmen wir an der stelle an: dass $0.01A$ sind, das bleibt, aber wir haben, den verstärkungsfaktor 500. das heisst, von dem strom. müssen wir - eine seltsame Idee. denn auch der, aber wir können in der schaltung gucken, im datenblatt. gucken: und da steht: I_{BE}. das ist die stromstärke Basis zu Emitter. und: trotzdem die ist wahrscheinlich bei: $20mA$ oder ähnlichem. oder sogar eben: nein, das kann ja nicht sein. oder doch? geteilt durch 500. ist zwar richtig. blos, selbst das in frage zu stellen. was ist denn: die obere spannung? 12V wahrscheinlich beim BC548. Blos, das ist nicht das Problem. wir machen das einfach. dann sehen wir. und gucken im datenblack. was wir als aufgabe brauchen ist das Timing Diagramm - teil vom Aktivitätsdiagramm.
- also: $5V$ an der Basis rein. und bei $0.01mA$ müssen wir es dahin schalten. jetzt in dieser aufgabe. und: das heisst: noch mal geteilt durch 500. so. das ist: wenn wir ein 500 Faches verstärkung haben, muss der Widerstand ein 500 faches sein, weil: der Strom und die Spannung sind ein 500tel. aber das jetzt nicht.

- $\frac{5V}{0.01mA} = 500\Omega$. das mal 500 sind $2.5k\Omega \cdot 100 = 250k\Omega$ an die Basis

jetzt beschriften wir das: und machen das an die Basis. und zeichnen.



und jetzt üben wir vier schaltungen: ich weiss das selber nicht, aber es gibt eine Basisschaltung, eine Emitterschaltung, eine Collectorschaltung.
fangen wir an:

1. Basisschaltung
2. Emitterschaltung
3. Collectorschaltung

ich nehme an, dass diese schaltungen ihren Ruf nur dem verdanken. dass es keine ernsthafte „logik“dahinter gibt. sie haben gesehen, ich habe versucht einen FET - Transistor - einen Feldeffekt Transistor selber her zu stellen. Er funktioniert, indem eine weitere Spannung mehr oder weniger einen Strom/Spannung erzeugt VCC - Voltage Current - Collector. Die Betohnung liegt auf Voltage und Current. doch trotz allem: das müsste meiner meinung nach mit einem normalen grossen Kristall gehen. den habe ich da oben bei dem Inder gekauft. es gibt in der Natur Kristalle und von Menschen gemachte. sie lassen sich mit Wasser selber machen. Kosmos.

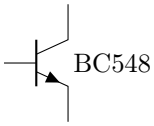
Denken wir an die Diode, dann wissen wir: Die Germanium Diode nimmt einfach Germanium und setzt darauf Metall. Vermutlich Aluminium. ich schaute in wikipedia. dort steht eine Abbilung eines Transistors. des ersten. sie ist vermutlich, dieser Transistor ziemlich gross. es wäre nicht erstaunlich. denn die Kondensatoren an sich sind eigentlich ziemlich gross. Ein Metall auf einem Strohmleitenden Metall müsste es möglich machen, dass wir einen FET herstellen.

gut zurück zu unserem PNP/NPN: BC 549, BC 548, es gibt weitere...542 glaube ich nach oben hin. haben Basis, Emitter, Collector. Doch solange die Grenzwerte nicht überschritten sind, liessen sie sich beliebig schalten, denn es interessiert das Bauteil an sich nicht. also tun wir erst mal exemplarisch, diese schaltungen hin.

das ist übung von circuitikz gleichzeitig! das müssen wir uns klar machen! doch im zweiten schritt geht es um weitere schaltungen. ein Widerstand am Emitter und GND, einer zwischen Collector und V_B. einer Zwischen Basis und Collector. Oder Spannungsteiler zwischen U_B und Basis und Emitter und Basis. Wir

lassen uns nicht durcheinander bringen. nur. wir müssen jetzt nicht alles wissen. es gibt namen und manche werte stimmen nicht. wir pinnen es mal hin. (1.) und zweitens: wir merken uns, was wir in CircuiTikZ geschrieben haben! ich zeige noch mal! die schaltungen und quelltext von gerade eben!

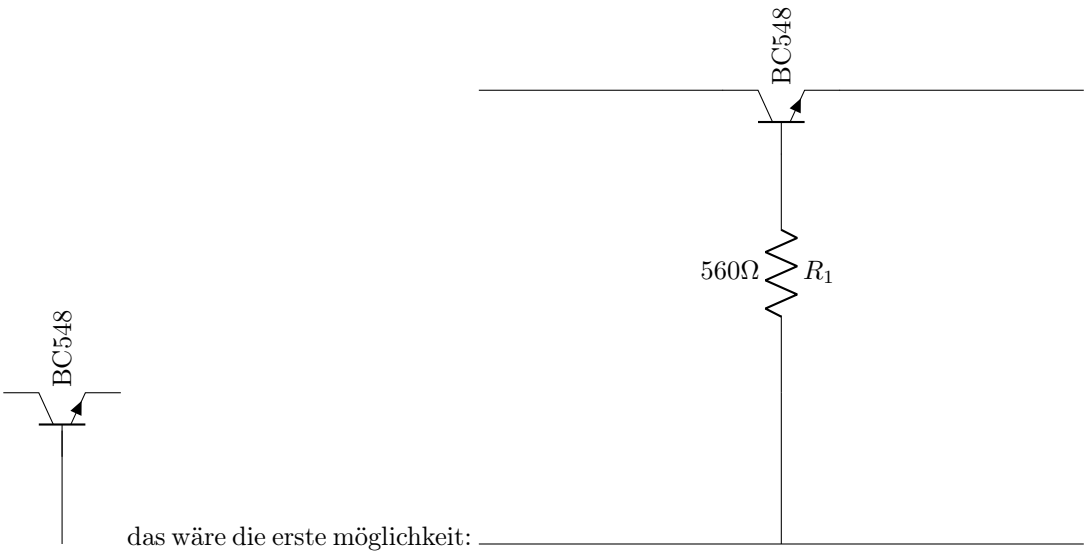
```
\begin{circuitikz}
    \draw (0,0) node[npn] (T1) {};
    \draw (T1.base) node[anchor=east] {B}
          (T1.collector) node[anchor=south] {C}
          (T1.emitter) node[anchor=north] {E};
    \draw (T1.collector) to[R, l=$R_C$, a=$330\Omega$] ++(0,2);
    \draw (T1.base) to[R, l=$R_B$, a=$250k\Omega$] ++(-2,0);
\end{circuitikz}
```



also:

so, unsere drei schaltungen:

1. Basisschaltung
2. Emitterschaltung
3. Collectorschaltung



das wäre die erste möglichkeit:

gut natürlich die weiteren schaltungen. wobei mir nicht klar ist, das schauen wir später ist bei der Emitterschaltung, der Verbraucher im Emitter - zwischen Basis Emitter. jetzt machen wir mal andere aufgaben, denn das sind nur übungsaufgaben.

das Periodensystem und das Timing-Diagramm
nun, und das müssen wir auch üben, das noch mit beiden Transistor Schaltungen und nicht vergessen

VDIV
HDIV

jetzt

VDIV	Voltage/Power/Ampere	$5V$	$2V$	$1V$	$0.5V = 500mV$	$0.2V = 200mV$	0.1
HDIV	Time/Seconds $5 \cdot t_1 \cdot s$	$2 \cdot t_1 \cdot s$	$1 \cdot t_1 \cdot s$	$0.5 \cdot t_1 \cdot s$	$0.2 \cdot t_1 \cdot s$	$0.1 \cdot t_1 \cdot s$	

passt jetzt hier nicht rein, die tabelle. nur: der unterschied t_1, ist ein schlechter faktor. schlechter name, für einen solchen Faktor.

und jetzt müssen ausrechnen, aber das machen wir jetzt nicht, das maximal Ding ist $\frac{1s}{10.000.000} = 10MHz$.
das rechne nicht aus. jetzt das Timing Diagramm + dazu mit atmega8. also, ich probiere - das hat noch ein externes Ding. als Triggereingang. jedenfalls: man kann das ja so machen, nachher das diagramm. ich muss mir jetzt schon wieder die namen merken. von dem alleine. jetzt pause. nachher noch das Periodensystem aus dem Kopf

18 nächster tag

H							He
Li	Be		C	N	O	Fl	Ne
Na	Mg	Al	Si	P	S	Cl	Ar
K	Ca					Br	Kr
		Fe	Co	Ni	Cu		

das war der erste versuch, ohne das vor augen zu haben.

H							He
Li	Be		C	N	O	Fl	Ne
Na	Mg	Al	Si	P	S	Cl	Ar
K	Ca	Fe	Co	Ni	Cu	Br	Kr

dann stimmt es nicht: weil Kalium K 19, und Eisen ist Fe 26. dann muss die erste tabelle richtig sein. und

H							He
Li	Be		C	N	O	Fl	Ne
Na	Mg	Al	Si	P	S	Cl	Ar
K	Ca					Br	Kr
		Fe	Co	Ni	Cu		

alternative es sind 9 spalten. weil: eigentlich kann: Cu nicht unter Brom? oder doch? es ist ja spaeter. edelgase und halogene. . . . . gut jedenfalls, das stimmt so nicht, weil da kommt Vanadium Chrom Mangan. und das muss vor das Eisen, Cobald, Nickel, Kupfer.

H							He
Li	Be		C	N	O	Fl	Ne
Na	Mg	Al	Si	P	S	Cl	Ar
K	Ca					Br	Kr
		Fe	Co	Ni	Cu		

trotzdem stimmt das nicht. warum?

weil: oder doch: weil H, ein aussenelektron. und He. hat zwei. das hat damit nichts zu tun. gut. dann ist das eine andere frage. dann noch mehr lücken. gucken nachher.

H								He
Li	Be			C	N	O	Fl	Ne
Na	Mg		Al	Si	P	S	Cl	Ar
K	Ca						Br	Kr
Va	Cr	Mn	Fe	Co	Ni	Cu	Id/Io	Xe

oder das Va, Cr, Mn gehört hinter Calcium. das heisst, ausserdem Titan Vanadium Chrom Mangan. Achso, Kalium Calcium Titan Vanadium Chrom Mangan. Also so:

H									He
Li	Be				C	N	O	Fl	Ne
Na	Mg			Al	Si	P	S	Cl	Ar
K	Ca			Ti	Va	Cr	Mn	Br	Kr
	Fe	Co	Ni	Cu: 29	Zn: 30	Ga: 31	Ge: 32	Id/Io	Xe

19 Begriffe Sortierung

drei neue begriffe:

1. Theorie
2. Modell
3. Methode

wir sortieren diese nicht ordentlich in einem baum, weil es für mehr nicht reicht. nur, für das was geht: also neue Übung! jeden tag die begriffe anordnen! nur lapidar!

1. Maschine
2. System

nächstes:

1. oder ...

oder ich habe doch eine einfachere Methode zum sortieren

Theorie
Modell
Methode

Theorem/Theorie

System
Maschine

Schaltnetz
Schaltwerk

System
Maschine
Automat

Automat
Algorithmus

Analyse von Schaltnetzen
Synthese von Schaltnetzen
Synthese von Schaltwerken
Analyse von Schaltwerken

Automat
Schaltwerk

deterministischer Automat
nicht deterministischer Automat
deterministischer Algorithmus
nicht deterministischer Algorithmus

terminiert
determiniert

Methode
Verfahren

Datenstrukturen
    Heap: Haufen
    Stapel: Stack
    Schlange: Queue

Komplexitaet

Zufall
Zufaelligkeit
Verteilung
Wahrscheinlichkeit

Zufallszahl
Zufallsexperiment

Experiment
Test
Darstellung
Simulation

Wahrscheinlichkeit
Zufallszahl
Zufallsexperiment

asymptotisch

symptotisch – Symptome
assymptotisch

Experiment
Methode
Test

Elementar
Additiv
Einfach
Zussammengesetzt
Adaptiv

linear

Kongruenz
Komplexitaet

Ergebnis
Ereignis
Fall

Ergebnis: Elementar Ergebnis
Elementare Datenstrukturen

Ergebnis, Ergebnismenge
S = {e0, e1, e2, ..., en}
das ist das was auftreten kann, beim wuerfel?
1, 2, 3, 4, 5, 6

Ereignis: das was vermutlich aufgetreten ist, oder wird:
E = {1,4}
ist vermutlich doch nicht aufgetreten, bezieht sich aber auf das Zufallsexperiment im Vollzug selbst

Ergebnis/Elementareignis?

vermutung: Ergebnisse ... ok

Blos: Ereignis A SUBSET von S
das heisst
A1 = {1,2}
A2 = {gerade/even}
A3 = {1, 3, 6}
usw. ...

egal:
Ergebnis

Ereignis
Fall

Fall:
    Zwei Wertpapier Fall (Markowitz)
    MehrwertPapier Fall (Markowitz)
assymptotischer Fall: Algorithmen

Theorie (Oekonomie)
    Neoliberalismus – das Modell des Modernen Kapitalismus
    Mikrooekonomie:
        Haushaltstheorie
        Preistheorie
        Produktionskostentheorie
        Marktversagen
    Finanzwissenschaft:
        Marktheorie...
    Theorie:
        Die Klassische Nationaloekonomie
            Adam Smith und Locke
        Der Monetarismus nach Milton Friedmann
        Der Keynesianismus
        Der Marxismus
Modelle:
    Kapitalismus vs. Sozialismus
    Freie Marktwirtschaft vs. Planwirtschaft

    Kapitalismus
    Sozialismus
    Freie Marktwirtschaft
    Planwirtschaft

Mathematik:
    Zufallszahlen
    sonst:
        Zufallsexperiment
        Das gesetz der grossen Zahlen
        Gleichverteilung: Laplace

    Zufaelligkeit
    Beliebigkeit
    Verteilung

Erwartungswert, Varianz, Standardabweichung

Varianz Var
Covarianz Cov

ok:

```
        Methode der linearen Kongruenz
        Methode der additiven Kongruenz
Methode/Test
        Test: Chi Quadrat Test
Laufzeit von Algorithmen
        N, NP
dann:
        Terminologie

Terminologie:
        Terminierung – geordnet beenden
        Kill – Toeten
        Abort: Abnormal Beenden
        Quit: Verlassen
        Hangup: auflegen
        Interrupt: unterbrechen

Theorie
Modell
Methode
Terminologie
Theorem
Test

Darstellung
Simulation
Automat, Algorithmus

Integral
Kurvenanpassung/Glaettung

Integral:
        symbolisch
        mit quadraten
                quadraturverfahren
                quadratur
                verfahren

        quadraturverfahren
        zusammgensetzte verfahren
        adaptive quadratur
kurvenanpassung:
        lagrange
        splines
        kleinste quadrate?

FFT (Kurvenanpassung)
        Elementare Einheitswurzeln

Polynom
```

```
        Interpolation
        Multiplikation
FFT:
        elementare Einheitswurzeln +
        Interpolation

Wurzeln und Quadrat:
        Integration +
        Kurvenanpassung

        interpolation:
                lagrange + splines

ok:
        Natuerliche Zahlen, Ganze Zahlen, Rationale Zahlen, Irrationale
        Reelle: Rationale + Irrationale

        Boolesche Algebra, Schaltalgebra:
                Boolean = {true, false}
                Schalt = {0,1}
        Polynome
        Matrizen
        Relationen (Karthesisches Produkt)
                Relationale Datenbanken
        Graphen: V - Vertices, Kanten E - Edges
                Paare: (Kanten E), Planare) oder? auch bei nicht planaren?
        Grosse Ganze Zahlen
        Zufallszahlen

        Fibonacci Zahlen

        Grosse Ganze Zahlen
                Addition...
        Matrizen
                Addition...
        Polynome
                Addition...

Prinzip
Modell
Theorie
```

also

$$\left(\begin{array}{c} n \\ k \end{array} \right)$$

$$\left(\begin{array}{c} n \\ k \end{array} \right) = \frac{n!}{k! \cdot (n-k)!} \tag{82}$$

und:

$$\left(\begin{array}{c} 4 \\ 2 \end{array} \right)$$

ok, wir berechnen, im kopf:

(4*3)*2*1 = 12*2 = 24

2! = 2

(4−2)! = 2

24/4 = 6

ok richtig mathematisch:

$$\left(\begin{array}{c} n \\ k \end{array} \right) = \frac{n!}{k! \cdot (n-k)!} \tag{83}$$

$$\left(\begin{array}{c} 4 \\ 2 \end{array} \right) = \frac{4!}{2! \cdot (4-2)!} \tag{84}$$

$$\frac{4!}{2! \cdot (4-2)!} = \frac{4 \cdot 3 \cdot 2 \cdot 1}{2 \cdot 2} \tag{85}$$

$$\frac{4 \cdot 3 \cdot 2 \cdot 1}{2 \cdot 2} = \frac{12 \cdot 2}{2 \cdot 2} \tag{86}$$

$$= = 6 \tag{87}$$

vermutlich. nun 3 aus 8

ach so: eines fehlt noch: prinzip! Prinzip, Modell, Theorie!
schon enthalten!

n!/(k!*(n–k)!)

8 chr 3

8!/(3!*(8−3)! =
8!/(3!*5!)

8*7*6*5*4*3*2*1

(8*7)*(6*5)*(4*3)*(2*1)

56*30*12*2
= 1680*24
= 20000 + 12000 + 1600 + 4000 + 2400 + 320

```
    20000
+ 12000
+   1600
+.  4000
+   2400
+     320
```

---

oder von vorne her:

$32000 + 5600 + 3720$
$= 32000 + 8000 + 1300 + 20$
$= 40000 + 1320 = 41320$

ok.

$5! = (5*4)*(3*2) = 20 * 6 = 120$

$3! = 6$

$41320/(6*120) = 41320/720 = 57{,}2888$

ok, ueberpruefung jetzt nicht.
oder doch

$8! = 8*7*\ldots*2*1 = 40320$

laut taschenrechner, also ein bisschen daneben.

$5! = 120$
$8! = 40320$

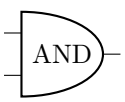
$40320 / 720 = 56$

$$\binom{8}{3} = 56 \tag{88}$$

vermutet.

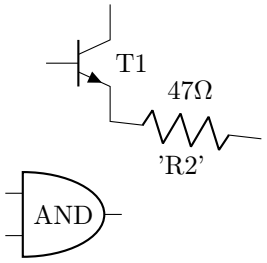
so im nächsten schritt, noch mal mit circuitkz die bauteile, danach timing-
R1



diagramm 


```
\begin{tikzpicture}
\node [and port] (and1) at (0,0) {AND};
\node [npn] (T1) at (0,2) {T1};
\node [R, a='R1', l=$47\Omega$] (R1) at (0,4) {R1};
\draw (T1.emitter) to [R, a='R2', l=$47\Omega$] (2,2);
\end{tikzpicture}
```

und, vielleicht gerade:
R1



jetzt ein paar Algorithmen.

20 BNF - beseitigung der Linksrekursion

laut google ai und auch sonst..., jetzt einfach benutzt. Sethi, Ulen, erneut kaufen + dijkstra + fernuni hagen - neues buch sethi ulen, weihnachten, addison wesley.

```
<expr>          ::= <term> <expr_prime>
<expr_prime>    ::= "+" <term> <expr_prime>
                  | "-" <term> <expr_prime>
                  | e

<term>          ::= <factor> <term_prime>
<term_prime>    ::= "*" <factor> <term_prime>
                  | "/" <factor> <term_prime>
                  | e

<factor>        ::= "(" <expr> ")"
                  | <number>
```

e entspricht epsilon. für mich ergibt das einen sinn. einfach zu merken: zwei mal

1. TERM EXPR2
2. TERM EXPR2

das heisst:

1. TERM EXPR2
2. + TERM EXPR2

ist aber fast das gleiche, wobei:

1. TERM EXPR2
2. + TERM EXPR2 | E

gut, anders ausgedrückt

$$\{A, A', \alpha, \beta\}$$

dabei gilt

- zu expr gehört das plus, grammatik ohne beseitigung der linksrekursion, jetzt gleich von mir
- indem falle aber mit term
- +term und term sind fast das gleiche, weil das + hier in diesem zusammenhang auftritt.

deswegen

```
<expr> ::= <expr> '+' <term> | <term>
<term> ::= <term> '*' <fact> | <fact>
<fact> ::= '('<expr>')' | <num>
```

ok,

$$\alpha \equiv +\text{term} \tag{89}$$
$$\beta \equiv \text{term} \tag{90}$$

so, der normale Ausdruck ist, oder der Normale Term

$$A \longrightarrow A\alpha|\beta \tag{91}$$

das heisst:

```
<expr> ::= <epxr> '+' <term> | <term>
<term> ::= <term> '*' <fact> | <fact>
```

$$\alpha \equiv \{+\text{term}, *\text{fact}\} \tag{92}$$
$$\beta \equiv \{\text{term}, \text{fact}\} \tag{93}$$

ok. und beseitigung der linksrekursion:

$$\{A, A', \alpha, \beta\}$$

$$A \longrightarrow A'\beta \tag{94}$$
$$A' \longrightarrow A'\alpha|\varepsilon \tag{95}$$
$$\tag{96}$$

aber eigentlich, sonst wieder linksrekursion:

$$A \longrightarrow A'\beta \tag{97}$$
$$A' \longrightarrow \alpha A'|\varepsilon \tag{98}$$
$$\tag{99}$$

gut nächster schritt, ein compiler:

```
// (o) david vajda
// 11/14/2026
// parsing - top-down - without beseitigung der L Recursion
```

```
int expr () {
        expr ();
        if (gettoken () == '+')
                term ();
}

int term () {
        term ();
        if (gettoken () == '*')
                fact ();
}

int fact () {
        int ch = gettoken ();
        if (ch == '('
                expr ();
                if (gettoken () != ')'
                        error ();
        }
        else if (isnum(gettoken ())
                //....
}
```

mit beseitigung der linksrekursion

```
// (o) david vajda
// 11/14/2026
// parsing - top-down - with beseitigung der L Recursion

int expr () {
        term ();
        expR ();
}
int expR () {
        if (gettoken () == '+') {
                term ();
                expR ();
        }
}
int term () {
        fact ();
        terM ();
}
int terM () {
        if (gettoken () == '*') {
                fact ();
                terM ();
        }
}
int fact () {
        int ch = gettoken ();
```

```
        if (ch == '(')
                expr ();
                if (gettoken () != ')'
                        error ();
        }
        else if (isnum(gettoken ())
                //....
}
```

als nächstes automat + schachfiguren + halt vorher! KMP!

20.1 Knuth-Morris-Pratt

Die Nuancen.

```
for (i = 0, j = 0, //...
for (i = 0, j = -1, //...

for (i = 0, j = 0; j < M && i < N; //...
for (i = 0, j = -1;  j < M; //...

for (i = 0, j = 0; j < M && i < N; i++, j++) //...
for (i = 0, j = -1;  j < M; i++, j++, next [i] = j) //...

// next [i] = j
// j = next [j]
```

jetzt das gesamte.

```
kmpsearch (char *a, char *p) {
        int i, j, M = strlen (p), N = strlen (a);

        kmpinit (p);

        for (i = 0, j = 0;  i < N && j < M;  i++, j++)
                while ((j >= 0) && (a [i] != p [j]))
                        j = next [j];
        if (j == M)
                return (i -M);
        else
                return i;
                // oder -1;
}

kmpinit (char *p) {
        int i, j, M = strlen (p);

        next [0] = -1;
```

```
        for (i = 0, j = -1;  j < M;  i++, j++, next[i] = j)
                while ((j >= 0) && (p[i] != p[j]))
                        j = next[j];
}
```

21 Automat

was ich weiss,

1. ich lerne die beispiele auswendig.

Beispiel:

```
0 1 2 3 4 5 6 7 8 9
10 states

Alpha Bravo Alpha Charlie Delta

Alpha Bravo Alpha
Alpha Bravo Charlie Delta

Pause Alpha Pause Bravo 2 x Pause
Pause Alpha Pause Bravo Pause Pause

x leer 88
C leer 88

und: 1: 22
3: 44
6: 77
7: 88
8: 99

31 und 61

state              0 1 2 3 4 5 6 7 8 9
ch [state]         A.   B.     A C D
next1[state].      1 2 3 4    6 7 8 9
next2[state].      1 2 1 4.   2 78 9
```

also, noch mal als richtige tabelle

state	0	1	2	3	4	5	6	7	8	9
ch [state]		A		B			A	C	D	
next1[state]	1	2	3	4	6	8	7	8	9	
next2[state]	1	2	1	4	2	8	7	8	9	

so weit so schön

```
deterministischer automat
nicht deterministischer automat
```

deterministischer algorithmus
nicht deterministischer algorithmus

Schlange
Stapel
Haufen

Schlange
Warteschlange
Doppelseitige Warteschlange

Stack: push, pop
Queue: put, get

deque: push, put, get

Nullzustand, Anfangszustand
Endzustand, Nachfolgezustand

Zustandsautomat
Mustererkennungsautomat
Automat
deterministischer Automat

Laufzeit O(N^2)

polynomiale Laufzeit

P: In polynomialer Laufzeit durch deterministische Algorithmen loesbar
NP: in polynomialer Laufzeit durch nichtdeterministische Algorithmen loesbar
NP–Vollstaendig: NP = P

Polynomial

N^2, N^3,

vermutlich

N^3 + N^2
vermutlich

also
O(N^3+N^2)

und:
P: polynomiale laufzeit ! durch deterministische Algorithmen

durchschnittlicher Fall
unguenstigster Fall
pathologischer Fall

2 Wert Papier Fall
mehr wert papier Fall

Automat:
Huellenbildung
Oder
Verkettung

Konstruktion des Automaten

Konstruktion des Automaten: Aneinanderkettung
Konstruktion des Automaten: Huellenbildung
Konstruktion des Automaten: Oder

Or/Concatenation/Closure.

Dabei konstruiert man den Automaten aus Teilautomaten!

wichtiger Begriff aus der Graphentheorie (Algorithmische Mathematik - Fernuni Hagen - Hochstättler, Nickel, Schulte, Andres) - 61411. und: hier untersuchung algorithen in C - Robert Sedgewick. Turing-Maschine - Turing, Computersysteme 1/2, Technische Informatik früher 1608/1609 Bähring, Schiffman, ... nicht turing aber Computerarchitektur, Turing Selber, ..., Knuth. Untersuchungen zu NP-Vollständigkeit. die Mathematiker der Graphentheorie anfangen bei Euler. Später andere. Kruskal, Kuratowski, Douglas, West, Harary, Heiratsproblem - John V. Atanasov, Loveplace, Charles Babbage, Zuse, Neumann, Harvard.

also Automat über Automat

1. Konstruktion des Automaten
2. Zustandsautomat
3. Mustererkennungsautomat

folgendes: die Konstruktion des Automaten:

1. Konstruktion des Automaten
   (a) Konstruktion des Automaten: verkettung
   (b) Konstruktion des Automaten: Oder
   (c) Konstruktion des Automaten: Hüllenbildung
2. Zustandsautomat
3. Mustererkennungsautomat

also, der Automat über den Automaten zusammensetzen. lauter teilautomaten über teilautomaten. ach ja, jetzt, eben an der stelle, was hier fehlte
Algorithmische Mathematik

```
Komponente
Teilgraph

Teilgraph
entspricht Teilautomat

der begriff Komponente ist extrem wichtig in der graphentheorie
Zusammengesetzter Graph und zwei unabhaengige Graphen ebenso

Vollstaendiger Graph
Komponente
Teilgraph

Zustandsdiagramm
Zustandsautomat
ok:

wenn ich Teilgraphen zusammensetze.
die untersuchung, wo sind teilgraphen

aber wo sind teilautomaten:
        Closure
        Concatenation
        Or

ok:
        Tiefensuche
        Breitensuche

das fuehrt zum mustererkennungsautomat
        Tiefensuche: Rekursion
        Breitensuche: Breath First Search

ist mit Q - wie Queue ist eine schlange
schlange - warteschlange
        doppelseite warteschlange:
                put, push, pop
queue: put,get
stack: push, pop

ok: kurz Breitensuche veranschaulichen
```

21.1 weil nicht passt auf seite

```
Q.put (r0)
r = Q.get ()
while !empty (r):
        w = neighbours (r)
        while v in w:
```

```
                Q.put (v)
r = Q.get ()
```

21.2 Breitensuche

```
Q.put (r0)
r = Q.get ()
while !empty (r):
        w = neighbours (r)
        while v in w:
                Q.put (v)
        r = Q.get ()
```

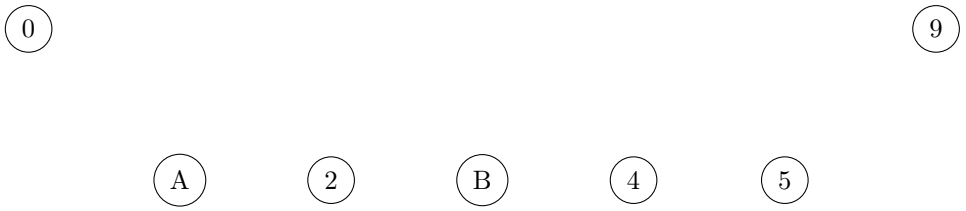
dieser algorithmus ist ähnlich dem, von algorithmischer mathematik, fernuni hagen, winfried hochstättler, andres, schulte, nickel. 61411. nur: er ist etwas umgeschrieben. ich vermute er entspricht dem in Robert Sedgewick, Algorithmen in C. und ist der allgemeine Algorithmus Breitensuche in Graphen.
ich habe ihn etwas umgeschrieben. damit er einfacher zu lesen ist und zwar aus dem kopf.

21.3 Der Mustererkennungsautomat/Nicht Deterministische Automat

state	0	1	2	3	4	5	6	7	8	9
ch [state]		A		B			A	C	D	
next1[state]	1	2	3	4	6	8	7	8	9	
next2[state]	1	2	1	4	2	8	7	8	9	

das ist jetzt nebenbei eine übung. da ich den algorithmus von





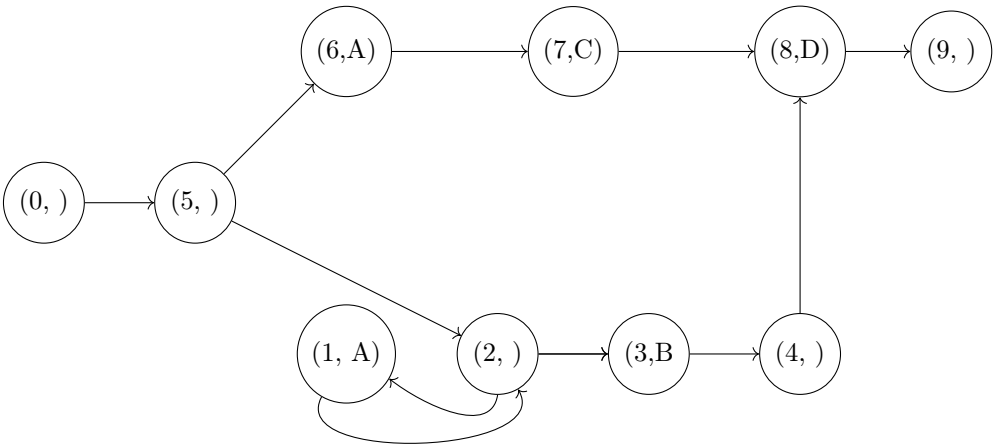
21.4 noch mal richtig!

ok zugegebener massen, das beispiel war auswendig! den automaten habe ich mittlerweile verstanden! vom programm-quelltext!

ich brauche noch mal das original!

state	0	1	2	3	4	5	6	7	8	9
ch [state]		A		B			A	C	D	
next1[state]	5	2	3	4	8	6	7	8	9	0
next2[state]	5	2	1	4	8	2	7	8	9	0

ok, anhand dessen jetzt, selbst den automat!



gut, die simulation des automaten ist recht easy. zunächst: computersysteme 1/2

1. analyse von schaltnetzen
2. synthese von schaltnetzen

aber

1. synthese von schaltschwerken
2. analyse von schaltwerken

schaltwerk und automat.

doch die ausführung selber (also, das was ich mit TTL Gattern tat - und knopf drücken per hand - LED blinken - zustand ändern) ist simulation.

die funktion dabei, bei einer linear eingegebenen zeichenkette sieht in etwa so aus

```
/* ... */

while (notempyty(deque)) {
        /* ... */
        if (a[j] == ch[state])
                put(state);
        else if (a[j] == ' ') {
                n1 = next1[state];
                n2 = next2[state];
                push(n1);
                if (n1 != n2)
                        push(n2);
        }
        state = get();
}
```

dazu kommt noch etwas mehr. nämlich die erhöhung des j im Text und das gleichzeitige einlesen eines zeichens. bei robert Sedgewick wurde das einlesen des

zeichens in den automat mit eingebaut. das einlesen des zeichens, ist dann etwas anderes, als der vergleich des Zeichens mit dem Text. oder der Automatenfolge. Das Einlesen und das Vergleichen mit einem Zustand mit Zeichen. aber: wenn das zeichen übereinstimmt.dann wird es vorne an die Schlange hinzugefuegt. wenn der Zustand wiederum kein Zeichen enthält, also Leerzustand, dann den Folgezustand 1x oder die Folgezustand 2x, wenn unterschiedlich hinten an die schlange dran, wie beim Stack.