

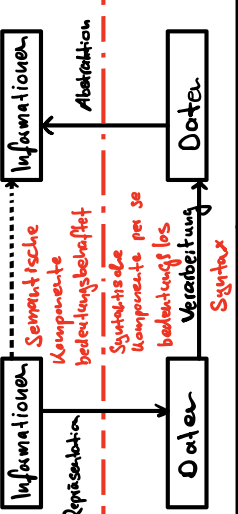
Turing Machine

Primitive Operationen:

- Rechts:** Kopf nach rechts
- Links:** Kopf nach links
- Drucken:** Symbol schreiben
- Lesen:** Symbol lesen
- Löschen:** Symbol löschen
- Nichts:** -

Diagramm

Semantik



Binär

Adressen: 10101010

Multiplikation: 10101010

Multiplikation: 10101010

Einzelkomplement: Bitflip

Zweierkomplement: Bitflip + 1

DEA vs. NEA

(c b a c) * c

Kürzester Weg zuerst C

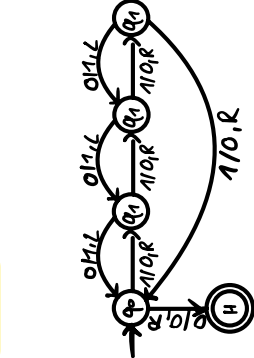
achal -> binär

Her -> binär

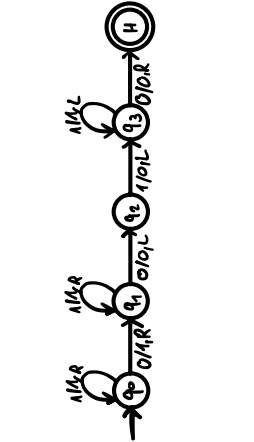
7	6	5	4	3	2	1	0
1111	1110	1101	1100	1011	1010	1001	1000

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
1111	1110	1101	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001	0000

Mod 4



Addition



Huffman - Encoding

1. Nach Häufigkeit sortieren
2. Knoten erstellen und mit Anzahl beschriften.
3. Neue Knoten erstellen aus Knotenanzahl.
4. Knoten markieren.
5. Bäume abgehen.

Rekursion

While

```

def summe(n):
    m = 0
    while n > 0:
        m += n
        n -= 1
    return m

```

Tail - Recursion

```

def summe(n):
    return summe_er(m, n)
def summe_er(m, n):
    if n == 0:
        return m
    return summe_er(m + n, n - 1)

```

Was macht Wolf:

Er löst Probleme.

Laufzeit Komplexität

- 1. Konstant
- log(n) — Logarithmisch
- log²(n) —
- sqrt(n) —
- n — Linear
- n · log(n) — Log-Linear
- n² — quadratisch
- n³ — kubisch
- 2ⁿ — Exponentiell

Sorts	Best	Avg	Worst	Speicher	Stabil
Bubble	n ²	n ²	n ²	1	Ja
Selection	n ²	n ²	n ²	1	best. ngt
Insertion	n	n ²	n ²	1	Ja
Quick	n log n	n log n	n ²	avg: log n best: n	best. ngt
Merge	n log n	n log n	n log n	n	Ja

Sortier Algorithmen

Selection Sort

1. Suche das kleinste Element
2. Sortiere es nach Links
3. Repeat

Quick Sort

1. Pivot wählen (Letztes Element)
2. Erstelle eine "kleiner" und eine "größer" Liste
3. Füge alle Elemente außer das Pivot-Element in die richtige Liste ein
4. Füge die linke-Liste, Pivot und rechte-Liste rekursiv zusammen.

Insertion Sort

1. Nimm das erste unsortierte Element
2. Setze es in den sortierten Teil der Liste ein (binary search)
3. Repeat

Merge Sort

1. Listen halbieren bis Länge 1 ist
2. Teillisten zusammenfügen zu sortierten Listen.
3. Fertig

achal -> binär

7	6	5	4	3	2	1	0
1111	1110	1101	1100	1011	1010	1001	1000

Her -> binär

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
1111	1110	1101	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001	0000