1 Abzählungen

Es gibt nur zwei Arten von Aufgaben:

- Anzahl Aufteilungen von einer Menge N von Kugeln in eine Menge R von Fächern
- Aus einer Menge N mit n Elementen sollen alle oder k Elemente ausgewählt werden

1.1 Anzahl Aufteilungen von einer Menge N von Kugeln in eine Menge R von Fächern

$\lvert N\rvert = n, \lvert R\rvert = r$	beliebig	injektiv	surjektiv	bijektiv
N unterscheidbar R unterscheidbar	r^n	$r \geq n : r^{\underline{n}}$ $r < n : 0$	$n \geq r : r! S_{n,r}$ $n < r : 0$	$r = n : n!$ $r \neq n : 0$
N nicht unterscheidbar R unterscheidbar	$\binom{r+n-1}{n} = \frac{r^{\overline{n}}}{n!}$	$r \geq n : \binom{r}{n}$ $r < n : 0$	$n \geq r : \binom{n-1}{r-1}$ $n < r : 0$	$r = n : 1$ $r \neq n : 0$
N unterscheidbar R nicht unterscheidbar	$\sum_{k=1}^{r} S_{n,k}$	$r \geq n : 1$ $r < n : 0$	$n \geq r : S_{n,r}$ $n < r : 0$	$r = n : 1$ $r \neq n : 0$
N nicht unterscheidbar R nicht unterscheidbar	$\sum_{k=1}^{r} P_{n,k}$	$r \geq n : 1$ $r < n : 0$	$n \geq r : P_{n,r}$ $n < r : 0$	$r = n : 1$ $r \neq n : 0$

1.2 Aus einer Menge N mit n Elementen sollen alle oder k Elemente ausgewählt werden

2 Codierung

2.1 Allgemeines

- Linearer (n, m)-Code C
- ***a*** **x** ***b*** **Matrix:** $n = b$ und $m = a$

2.2 Wichtige Formeln

- **Blocklänge:** n
- **Linear unabhängige Wörter/Dimension des Unterrraums** C**:** m
- **Anzahl Codewörter:** $|C| = q^m$, wobei q Anzahl Elemente in C
- **Anzahl Wörter in Standardfeld:** q^n
- **Anzahl Wörter in Syndromtabelle:** q^{n-m}
- **Hamming Code:** $n = \dfrac{q^{n-m} - 1}{q - 1}$
- **Schätzen der Codedistanz (Singleton-Schranke):** $d(C) \leq n - m + 1$
- **Fehlererkennend:** $(n - m)$
- **Fehlerkorrigierend:** $\left\lfloor \dfrac{(n - m)}{2} \right\rfloor$
- **t-fehlererkennend:** $d(C) \leq t + 1$, das ausgewählte t ist, wie viel Fehler der Code erkennt
- **t-fehlerkorrigierend:** $d(C) \leq 2t{+}1$, das ausgewählte t ist, wie viele Fehler der Code korrigiert

2.3 Codewörter sind gegeben

n und m bestimmen

$$n = \text{Länge der Codewörter} \Rightarrow n = 5$$

00000
01101
10111
11010

$$m = \text{Dimension} \Rightarrow m = 2$$

00000
01101
10111
11010

Hamming-Distanz/Code-Distanz/$d(c)$

- Hemming-Distanz ist die minimale Änderung des Gewichts
- Wird mit $d(C)$ bezeichnet

Beispiel:

- Anzahl Einsen in Codewörter Zählen
- $0 \ldots 0$ wird dabei nicht beachtet

01101 $\rightarrow$ Gewicht: 3
10111 $\rightarrow$ Gewicht: 4
11010 $\rightarrow$ Gewicht: 3

- Gewicht der Codewörter vergleichen und minimalstes auswählen

$\Rightarrow$ $d(C) = 3$, da es minimal ist

t-fehlererkennend

t-fehlerkorrigierend

t-ausfällekorrigierend

Kanonische Generatormatrix

- Definition: Aus der Generatormatrix kann man alle möglichen Codewörter der Sprache erzeugen.
- Falls die Generatormatrix gegeben ist, lässt sich die die kanonische Generatormatrix durch das Anwenden des Gaußschen Verfahrens auf die Generatormatrix erstellen.
- Größe: $(m \times n)$
- Aufbau: $G = \begin{pmatrix} E & G' \end{pmatrix}$
  - E: Einheitsmatrix
  - G': Linear unabhängiger Rest aus Codewörtern

$$G = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

Kanonische Kontrollmatrix

- Definition: Erfüllt ein Wort $wort \cdot kontrollmatrix = 0$ ist es ein richtiges Codewort.
- Größe: $n \times (n - m)$
- Aufbau: $H = \begin{pmatrix} -G \\ E \end{pmatrix}$
  - E: Einheitsmatrix
  - $-G$: Linear unabhängiger Rest aus Codewörtern

$$H = \begin{pmatrix} -1 & -1 & -1 \\ -1 & -0 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Modulo mit negativen Zahlen:

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9
$\mathbb{Z}_2$									0	1	[0	1]								
$\mathbb{Z}_3$								0	1	2	[0	1	2]							
$\mathbb{Z}_4$							0	1	2	3	[0	1	2	3]						
$\mathbb{Z}_5$						0	1	2	3	4	[0	1	2	3	4]					
$\mathbb{Z}_6$					0	1	2	3	4	5	[0	1	2	3	4	5]				
$\mathbb{Z}_7$				0	1	2	3	4	5	6	[0	1	2	3	4	5	6]			
$\mathbb{Z}_8$			0	1	2	3	4	5	6	7	[0	1	2	3	4	5	6	7]		
$\mathbb{Z}_9$		0	1	2	3	4	5	6	7	8	[0	1	2	3	4	5	6	7	8]	
$\mathbb{Z}_{10}$	0	1	2	3	4	5	6	7	8	9	[0	1	2	3	4	5	6	7	8	9]

2.4 Syndromtabelle

- Definition: Identifiziert und korrigiert Fehler in Codewörter durch Vergleich mit erwarteten Werten.
- Anzahl Zeilen: q^{n-m}
- Anzahl Spalten: n

Beispiel: (7×3)-Kontrollmatrix, $n = 7, m = 4$

$$H = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

a	$S = a \cdot H$
0000000	000
1000000	111
0100000	101
0010000	011
0001000	110
0000100	100
0000010	010
0000001	001

- Anzahl Zeilen: $2^{7-4} = 8$
- Anzahl Spalten: 7
- Wenn über 0000001 hinaus geht egal ob 1000001, 1100000, ...

1. Überprüfen ob es ein empfangenes Codewort fehlerfrei ist($wort \cdot kontrollmatrix = 0$)
   - Empfangenes Wort: $y = 1010010$

$$1010010 \cdot \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = 110 \neq 0 \Rightarrow \text{Fehler im empfangenen Codewort}$$

2. Codewort durch Syndromtabelle korrigieren
   - Klassenanführer von 110 aus Syndromtabelle ablesen: $a = 0001000$
   - $empfangenes\ Codewort - Klassenanfuehrer = korrigiertes\ Codewort$

$$\begin{array}{ccc} y & & 1010010 \\ a & - & 0001000 \\ \hline & & 1011010 \end{array}$$

   - korrigiertes Codewort: 1011010

3. Nachricht extrahieren
   - Letzte Stellen des korrigierten Codewort entfernen, um die Nachricht zu erhalten (Anzahl entfernte Stellen entspricht Länge des Syndrom).
   - Nachricht: 1011~~010~~ = 1011

2.5 Standardfeld

2.6 Kontrollmatrix zu Generatormatrix umwandeln

$$H = \begin{pmatrix} 2 & 0 \\ 1 & 1 \\ 0 & 1 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \end{pmatrix}^T$$

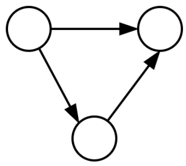
3 Graphentheorie

3.1 Grundbegriffe

$G(V, E)$, V = Vertices = Knoten, E = Edges = Kanten

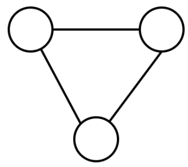
Gerichteter Graph

- Kante geht nur in *eine* Richtung



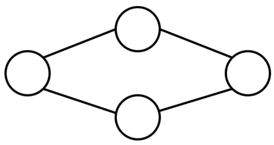
Ungerichteter Graph

- Kante geht nur in *beide* Richtung



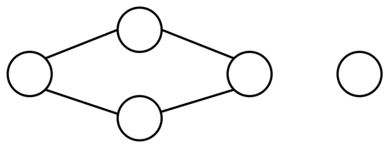
Zusammenhängender Graph

- Verbindung von einem Knotenpunkt zu allen anderen
- Verbindungen müssen nicht direkt sein
- Gibt keine isolierten Knoten



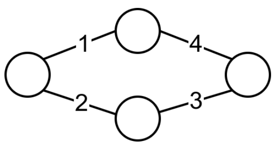
Nichtzusammenhängender Graph

- Wenn ein Knoten isoliert ist und keine Verbindung herrscht



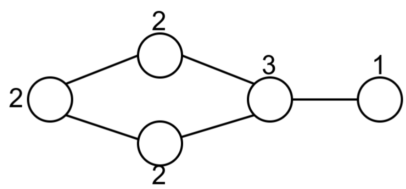
Gewichteter Graph

- Die Kanten bekommen Gewichte zugewiesen



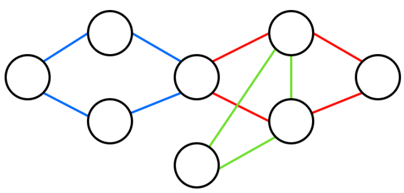
Knotengrad

- Anzahl Kanten die von einem Knoten ausgehen



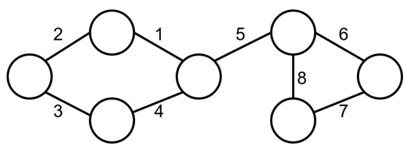
Euler-Zyklus/Eulersch

- Anfangszustand = Endzustand
- Alle Knoten haben einen geraden Grad



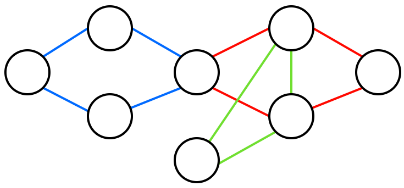
Euler-Pfad

- Jede Kante wird genau einmal durchlaufen
- Keine Kante wird mehrmals durchlaufen
- Startknoten muss nicht gleich Endknoten sein
- Genau zwei Knoten haben einen ungeraden Grad



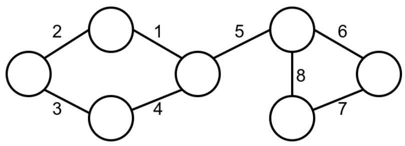
Hamilton-Zyklus

- Anfangszustand = Endzustand
- Alle Knoten geraden Grads



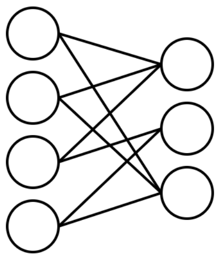
Hamilton-Pfad

- Jede Kante wird genau einmal durchlaufen
- Keine Kante wird mehrmals durchlaufen
- Startknoten muss nicht gleich Endknoten sein



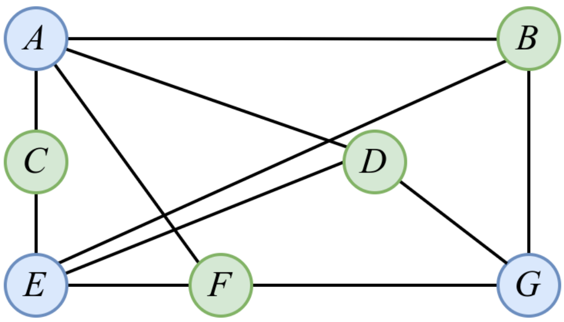
Bipartit

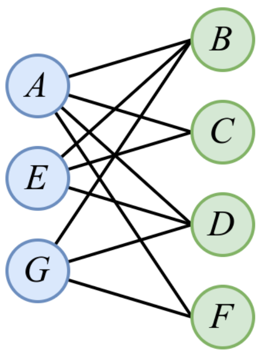
- Man kann alle Knoten in zwei (disjunkte) Teilmengen/Gruppen aufteilen
- Jede Verbindung geht dabei von einer Teilmenge/Gruppe in die andere
- Es darf allerdings keine Verbindung der Knoten innerhalb der eigenen Teilmenge/Gruppe geben



Beispiel: Zeigen ob ein Graph bipartit ist

- Beliebigen Knoten auswählen und färben (z.B blau).
- Alle benachbarte Knoten mit einer anderen Farbe färben (z.B grün)
- Widerholen: Alle Knoten die an einen blauen Knoten angrenzen, grün und alle Knoten, die an einen grünen Knoten angrenzen, blau färben.
- Prüfung: Wenn man auf einen Knoten stößt, der bereits gefärbt ist und dessen Farbe sich von der Farbe unterscheidet, die man ihm zuweisen möchten, dann ist der Graph *nicht bipartit*. Wenn man alle Knoten ohne solche Konflikte färben konnte, ist der Graph *bipartit*.
- Die blauen Knoten sind am Ende die eine Teilmenge/Gruppe und die grünen Knoten sind am Ende die andere Teilmenge/Gruppe





Planar

- Ein Graph ist planar, wenn man ihn so zeichnen kann, dass sich seine Kanten nirgendwo kreuzen.

(Nicht) Planarität prüfen:

1 Methode:

- $n = |V|$ (Anzahl Knoten) und $m = |E|$ (Anzahl Kanten) bestimmen
- Hat der Graph einen Zyklus der Länge 3?

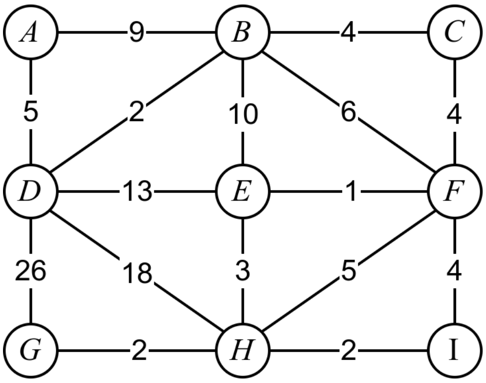
  → **Ja:** $m \leq 3n - 6$

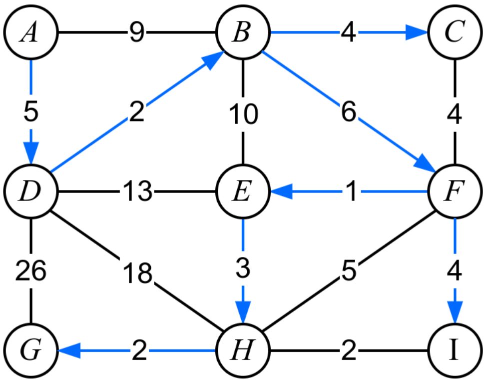
  → **Nein:** $m \leq 2n - 4$

  - Gilt $m \nleq 3n - 6$ oder $m \nleq 2n - 4$ ist der Graph *nicht planar*
  - **Achtung:** Die Formel macht nur eine Aussage darüber, ob der Graph nicht planar ist, aber nicht, ob ein Graph planar ist. Gilt also $m \leq 3n - 6$ oder $m \leq 2n - 4$ heißt es nicht, dass der Graph planar ist. Man kann also nur die Nichtplanarität überprüfen.

3.2 Kürzester Weg

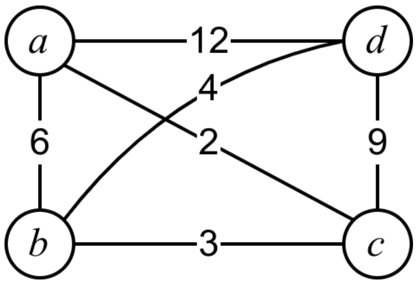
Dijkstra





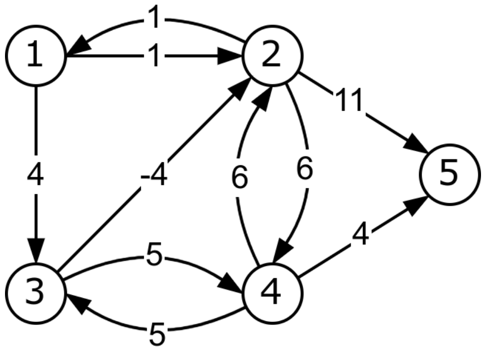
A	B	C	D	E	F	G	H	I	S
0(A)									A
0(A)	9(A)		5(D)						A, D
0(A)	7(D)		5(D)	18(E)		31(D)	23(D)		A, D, B
0(A)	7(D)	11(B)	5(D)	17(E)	13(B)	31(D)	23(D)		A, D, B, C
0(A)	7(D)	11(B)	5(D)	17(E)	13(B)	31(D)	23(D)		A, D, B, C, F
0(A)	7(D)	11(B)	5(D)	14(F)	13(B)	31(D)	18(F)	17(F)	A, D, B, C, F, E
0(A)	7(D)	11(B)	5(D)	14(F)	13(B)	31(D)	17(E)	17(F)	A, D, B, C, F, E, H
0(A)	7(D)	11(B)	5(D)	14(F)	13(B)	31(D)	17(E)	17(F)	A, D, B, C, F, E, H, I
0(A)	7(D)	11(B)	5(D)	14(F)	13(B)	31(D)	17(E)	17(F)	A, D, B, C, F, E, H, I, G

Floyd



	i	a	b	c	d	a	b	c	d
	a	0	6	2	12	a	a	a	a
	b	6	0	3	4	b	b	b	b
	c	2	3	0	9	c	c	c	c
	d	12	4	9	0	d	d	d	d
1	a	0	6	2	12	a	a	a	a
1	b	6	0	3	4	b	b	b	b
1	c	2	3	0	9	c	c	c	c
1	d	12	4	9	0	d	d	d	d
2	a	0	6	2	10	a	a	a	b
2	b	6	0	3	4	b	b	b	b
2	c	2	3	0	7	c	c	c	b
2	d	10	4	7	0	b	d	b	d
3	a	0	5	2	9	a	c	a	b
3	b	5	0	3	4	c	b	b	b
3	c	2	3	0	7	c	c	c	b
3	d	9	4	7	0	c	d	b	d
4	a	0	5	2	9	a	c	a	b
4	b	5	0	3	4	c	b	b	b
4	c	2	3	0	7	c	c	c	b
4	d	9	4	7	0	c	d	b	d

FIFO



D(s,1)	D(s,2)	D(s,3)	D(s,4)	D(s,5)	R(1)	R(2)	R(3)	R(4)	R(5)	S
0										1
0	1	4				1	1			$2 < 3$
0	1	4	7	12		1	1	2	2	$3 < 4 < 5$
0	0	4	7	12		3	1	2	2	$4 < 5 < 2$
0	0	4	6	11		3	1	2	4	$5 < 2$
0	0	4	6	11		3	1	2	4	2
0	0	4	6	10		3	1	2	4	4
0	0	4	6	10		3	1	2	4	5
0	0	4	6	10		3	1	2	4	