- Ein Weg ⇒ ein Pfad (Abkürzungen weglassen)

Wenn v_i zum ersten Mal besucht wird, springe an den Ort im Weg, wo v_i zum letzten Mal auftaucht.

- $\sum \deg(v) = 2|E|$ | $\sum \deg^+(v) = \sum \deg^-(v) = |A|$
- # Knoten mit $\deg(v) \equiv_2 1$ ist gerade
  Da $\sum \deg(v) = 2|E|$ und $\sum \text{evendeg}(v) + \sum \text{odddeg}(v) = 2|E|$
- $\exists x,y$ mit $\deg(x) \le \frac{2|E|}{N}$ und $\deg(y) \ge \frac{2|E|}{N}$

Die Länge des Wegs :≡ # Kanten, also Länge $(v_1,\dots,v_k)$ ist $k-1$

$\binom{V}{2}$:= Menge aller Paare aus V, sodass $x \neq y$

$\binom{V}{k}$:= Menge aller k-tupel aus V, ohne Duplikate
Menge aller Pfade ist eine Teilmenge dieses Graphen.

K_n :≡ Kompletter Graph mit n Knoten (k-regulär)

C_n :≡ Kreisgraph mit n Knoten (2-regulär)

Q_d :≡ Hyperwürfel von Grad $d \equiv \{0,1\}^d$
Also 010 ist ein Knoten und ist verbunden mit allen anderen Knoten. (d-regulär)

Multigraph :≡ Graph mit Schleifen und Mehrfachkanten.

$N_G(v) :\equiv \{u \in V : (u,v) \in E\}$ Nachbarschaft von v.

$\deg(v) = |N_G(v)|$ = Grad von v.

k-regulär :≡ $\forall v \in V: \deg(v) = k$

Zwei Knoten heissen adjazent, falls Kante existiert.

Ein Knoten und eine Kante heissen inzident, falls der Knoten ein Endknoten ist.

Schwacher Teilgraph :≡ $V_H \subseteq V_G$

Induzierter Teilgraph :≡ Knoten alleine definieren Teilgraphen, alle Kanten zwischen den Knoten bleiben erhalten.
Notation: $H = G[V_H] = E_G \cap \binom{V_H}{2}$

$W := \langle v_1, v_2, \dots, v_k \rangle$:≡ Weg $\{v_1, v_2\} \in W$

Zyklus :≡ Weg mit $v_1 = v_k$

Kreis :≡ Pfad von $v_1, \dots, v_{k-1}$ und $v_k = v_1$

$[n] :\equiv \{1, \dots, n\}$

Topologische Sortierungen

Compute indegree für jeden Knoten.
Dann subtrahiere für jede ausgehende Kante die Nachbarn.

Äquivalente Baumaxiome
- G ist zusammenhängend ∧ kreisfrei.
- G ist zusammenhängend ∧ $|E| = |V| - 1$
- G ist kreisfrei ∧ $|E| = |V| - 1$
- $\forall x \neq y \in V: \exists!$ x-y Pfad

Baum-Extrahierung (Spannbaum Kreierung)

Finde einen beliebigen Zyklus und entferne eine beliebige Kante e dieses Zyklus.

Offensichtlich ist $G \setminus \{e\}$ zusammenhängend, da jeder Weg der über e geht den Zyklus als Umweg nehmen kann.

Minimaler Spannbaum

$E_T \subseteq E$ und $\sum_{e \in E_T} c(e) \overset{!}{=} \min$

Mind. 1 Kante eines Schnitts gehört zu einem Spannbaum.

$E(S, V \setminus S)$ → Alle Knoten zwischen S und $V \setminus S$

$c(\hat{e}) = \min\{c(e) \mid e \in E(S, V \setminus S)\}$

a) Für jedes $\hat{e}$ gibt es ein minim. Spannbaum der $\hat{e}$ enthält. (Einfach)

b) Gibt es nur ein $\hat{e}$, ist $\hat{e}$ in allen minim. Spannbäumen enthalten.

Betrachten wir einzeln jeden Spannbaum.
1. $\hat{e} \in T$ □
2. $\hat{e} \notin T$
Falls $\hat{e} \in T$ müsste $\hat{e} \cup T$ ein Kreis bilden.
Also betrachten wir $T' = T \cup \hat{e} \setminus e'$ (wo e' eine andere Kante im Schnitt nach a))
Da $c(T') = c(T) + \hat{e} - e'$, und $\hat{e} \le e' \Rightarrow \hat{e} = e'$

Gegeben ein Kreis mit $e^* = \max(C)$

a) Es gibt einen Spannbaum der e^* nicht enthält.

b) $c(e^*) > c(e')$ für alle $e' \in C$, so ist e^* in keinem Spannbaum.

Für jeden min. Spannbaum T:
1. Ist $c^* \notin T$ □
2. Ist $c^* \in T$, realisieren wir, dass $T \setminus e^*$ nicht zusammenhängend ist.
Also $\exists e'$ im Kreis, der beide Komponente verbinden würde. Dieser Baum $T \setminus e^* \cup e'$ wär aber mind. gleich günstig (a) und strikt günstiger (b

Blaue Regel: In einem Schnitt ohne blaue Kanten darf einer der kleinsten Kanten blau gefärbt werden.

Rote Regel: In einem Kreis ohne rote Kanten darf einer der grössten Kanten rot gefärbt werden.

Keine Kante lässt sich färben ⇔ Alle Kanten sind gefärbt.

Beweis Blau: Da $|E| = n-1$, können wir so lange eine Kante blau machen bis kein Knoten mit inzidenter Kante gibt.

PRIM: $O(\min(V^2 + |E|, |E| \log V))$ oder $O(|V| \log |V| + |E|)$ mit Fib-Heap

$\sigma[x]$ = Kleinste entdeckte Kante von S nach x. pred[x] = Vorgänger dieser entdeckten Kante

Zuerst $\sigma[x] = \begin{cases} c(s,x), & (s,x) \in E \\ \infty, & \text{else} \end{cases}$ pred[x] = s v undef.

Insert$(Q, x, \underset{\text{key}}{\sigma[x]})$ $\forall x \in V \setminus \{s\}$, also $|V| \cdot \log(V)$, or $|V| \cdot 1$

while$(Q \neq \emptyset)$ $\hat{x} \in$ ExtractMin(Q) also $|V| \cdot \log(V)$ or $|V| \cdot |V|$
for$(x \in N(\hat{x}))$ if$(c(\hat{x}, x) < \sigma[x])$
Decrease Key$(Q, x, \sigma[x], c(\hat{x}, x))$; $\sigma[x] \leftarrow c(\hat{x}, x)$; $|E| \cdot \log(V)$ or $|E| \cdot 1$

KRUSKAL: $O(|E| \log |E|) = O(|E| \log |V|)$ → Sortieren
Insert: $O(1)$, Find: $O(1)$, X-Unions: $O(x \log(x))$ der Kanten [illegible]
$\alpha(V) \cdot E$ mit path compr. ∧ union-by-size

Borůvka Jeder Knoten formt Wald. ∀ Komponente Füge billigste von $C \to V \setminus C$ hinzu.
Parallelisierbar

∃ Eulertour :⇔ ∃ Geschlossener Weg (Zyklus) ⇔ Eulersch

MST kommt kein Kantengewicht doppelt vor
$\Rightarrow$ MST eindeutig

Beweis: Für jeden Schnitt gilt, dass wenn $e \in E(S, V\setminus S)$ kleineres Kantengewicht hat als der Rest, dann ist $e \in MST$

Lös 1:
Jeder Knoten kann man als seperaten Schnitt betrachten und dessen Kanten dazufügen. Wiederhole mit Schnitt auf Komponenten
Lös 2: $\forall e \in T$ gilt, dass $T\setminus\{e\} = A \oplus B$. Der Schnitt dazw. enthält nur eine minimale Kante.

$$\mathbb{E}[\#\text{ Kanten über Schnitt}] = \underbrace{\frac{2^{n-1}}{2^n}\cdot\frac{2^{n-1}}{2^n}\cdot 2}_{\text{Prob. Kante in } u \text{ und in } v}\cdot m = \frac{1}{2}m$$

Maximale Grösse eines Schnittes $\geq \frac{1}{2}m$

Erw. Anzahl Mädchen vermathören bis Junge: $\underbrace{\frac{1}{2}\cdot 1+\frac{1}{4}\cdot 1+\frac{1}{8}\cdot 1 \ldots}_{=1}$

Alice mit Münze gewinnt mit Wahrsch. $\frac{1}{2^i}(1-p)^{i-1}$ in Runde i.

Bob mit Münze $1/2 < p < 1$ mit Wahrsch. $(1-p)^{i-1}p\frac{1}{2^i}$ in Runde i.

$$\sum_{i=1}^{\infty}\frac{1}{2^i}(1-p)^{i-1} = \frac{1}{2}\sum_{i=1}^{\infty}\left(\frac{1-p}{2}\right)^{i-1} = \frac{1}{2}\frac{1}{1-\frac{1-p}{2}} = \frac{1}{\frac{1+p}{2}}\frac{1}{2} = \frac{1}{1+p}$$

Beispiel Graph mit min. Grad $|V|/2-1$ ohne Hamiltonkreis (folgt aus Paritätsargument) für $|V|$ ungerade.
bzw. für $|V|$ gerade.

Maximales Matching $=1$

Coupon-Collector letzte 11: $\frac{1}{\frac{11}{n}}+\frac{1}{\frac{10}{n}}+\cdots+\frac{1}{\frac{1}{n}}$

$$\sum_{i=1}^{11}\frac{n}{i} = n\sum_{i=1}^{11}\frac{1}{i} - n\ln(n)+\gamma+O\left(\frac{1}{n}\right)$$

$$\|s_1v_1+s_2v_2+\cdots+s_nv_n\|^2 =:$$

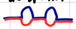
$$\Pr[|X-\mu|\geq 0.03] \leq \frac{\frac{1}{4n}}{(0.03)^2}$$

$$\Pr[|X-\mu| < 0.03] \geq 1-\frac{\frac{1}{4n}}{\frac{27}{1000}}$$

Zusammenhängend $\wedge$ kreisfrei $\Rightarrow |E|=|V|-1$
$N=1$: o
$N\geq 2$: Mind. 1 Kante, sonst nicht zusammenhängend. Max 1. Kante, da sonst $v_i \neq v_j$ verbunden über $v_i, v_a, \ldots, v_k, v_j$ mit den neuen Kanten einen Kreis bilden würden. $\square$

Zusammenhängend $\wedge$ $|E|=|V|-1 \Rightarrow$ kreisfrei
Per Widerspruch: Angenommen $\exists$ Kreis $C=(v_a, v_e, \ldots, v_k, v_a)$ mit k Kanten. Falls alle Knoten im Kreis $|E|\neq|V|-1$. Sonst entfernen wir die anderen Knoten nacheinander bis zum Kreis (bleibt Baum). $\square$

$\forall x,y \in V\ \exists! x,y$ Pfad $\Rightarrow$ zusammenhängend $\wedge$ kreisfrei
Falls Kreis hätte gäbe es 2 Pfade von u nach v im Kreis.
Da ein Pfad existiert, muss der Baum zusammenhängend sein. $\square$

Zusammenhängend $\wedge$ kreisfrei $\Rightarrow \exists! x,y$ Pfad $(\forall x,y \in V)$
Es existiert mind. ein Pfad von x nach y (zusammenhängend).
Gäbe es 2 Pfade müssen die Pfade irgendwann aufsplitten und wieder zusammenkommen. Das wäre aber ein Kreis:  $\square$

Baum generieren mit BFS, DFS oder in dem man eine Kante aus beliebigem Kreis entfernt.
(Kante entfernen erhält zusammenhang über Kreis für alle Wege mit u-v.)

Einseitiger Fehler:

Primzahlalgorithmus: $O(\log(n))$, Primzahl = Immer, Keine Primzahl = $3/4$

$A(I,R)$:= Ausgabe vom randomisierten Algorithmus.

Korrektheit hängt von Eingabe I $\wedge$ Zufallsvariable R ab.

$\Pr_R[A(I,R)$ ist korrekt$] \geq 1-\varepsilon$

Laufzeit: $\mathbb{E}[\text{Laufzeit}] = f(|I|)$ (Monte-Carlo)

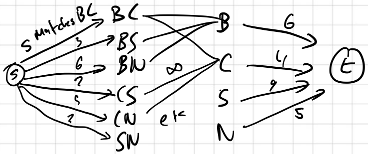
oder $\Pr_R[\text{Laufzeit} \leq f(n,\varepsilon)] \geq 1-\varepsilon$ (Las Vegas)

Las Vegas Algorithmus kann man in Monte-Carlo verwandeln indem man

$\Pr[X\leq 5] \leq 1/4$ falsch, da $\frac{85}{100}\cdot 0+\frac{1}{100}\cdot 100 = 1$

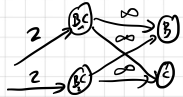
$$f_X(x) = \begin{cases} 1-1/n, & x=n \\ 1/n, & x=-n(n-1) \\ 0, & \text{sonst}\end{cases}$$

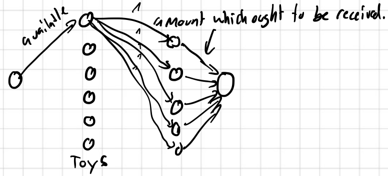
Mannschaftsspiel: Würde Puma immer gewinnen, hätten sie 12 Punkte. Max. Punkte für andere $= M = 12$ - haben sie schon.



(S hat schon 9 gewonnen.)

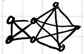
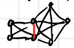
Schwereres Max Flow mit unentschieden.





$T(G)$:= # Untersch. spanning trees
$T(G) = T(G/e)+T(G\setminus e)$
Falls $e \in T$

$T(K_n) = n^{n-2}$, $T(K_n\setminus e) = (n-2)n^{n-3}$, $T(K_n/e) = 2n^{n-3}$
$T(K_n \ominus K_m) \overset{!}{=} T(K_n\setminus e)T(K_m/e)+T(K_n/e)T(K_m\setminus e)$

$K_4 \ominus K_5 =$  $K_4 \oplus K_5 =$ 

Coupon Collector: $n\log(n)+O(n) = \mathbb{E}[X]$

$\Pr[X \geq 100\, n\log(n)] \leq \frac{1}{100}$

$\Pr[|X-\mathbb{E}[X]| \geq 10] \leq \frac{n^2\frac{\pi^2}{6}}{10^2}$

$$\frac{\deg(u)}{2|E|}\cdot\frac{1}{\deg(u)}+\frac{\deg(v)}{2|E|}\cdot\frac{1}{\deg(v)} = \frac{1}{|E|}$$

Quickselect: Durchschn. Abstand vom grösseren Teil ist uniform zwischen $[\frac{1}{2}n, n]$, also $\mathbb{E}[X] = 3/4\, n$

Aus Linearität des Erwartungswerts folgt $\sum_{i=0}^{n}\left(\frac{3}{4}\right)^i n \leq n\sum_{i=0}^{\infty}\left(\frac{3}{4}\right)^i = 4n \in O(n)$

Zuerst ziehe $X \leftarrow \text{uniform}(1,\dots,n)$ $Y \leftarrow \text{uniform}(1,\dots,X)$

$$\Pr[X=x, Y=y] = \Pr[Y=y \mid X=x] \cdot \Pr[X=x] = \frac{1}{x} \cdot \frac{1}{n}$$

$$\Pr[Y=y \mid X=x] = \begin{cases} \frac{1}{x}, & y \leq x \\ 0, & \text{sonst} \end{cases}$$

$$\Pr[Y=y] = \sum_{x \in \Omega} \Pr[X=x, Y=y] = \sum_{x=y}^{n} \frac{1}{x} \cdot \frac{1}{n} = \frac{1}{n}(H_n - H_y) \approx \frac{\log(n) - \log(y)}{n}$$

$x=y \to$ sonst 0

$$\Pr[X=x \mid Y=y] = \frac{\Pr(X=x, Y=y)}{\Pr[Y=y]} = \frac{\frac{1}{x} \cdot \frac{1}{n}}{\frac{1}{n} \cdot (H_n - H_y)} \approx \frac{1}{x(\log(n) - \log(y) + O(\frac{1}{n}))}$$

f) Nicht unabhängig. $n=2$ $X \leftarrow \text{uniform}(1,2)$ $Y \leftarrow (1, X)$

$$\Pr[X=1, Y=1] = \frac{1}{2} \neq 1 = \Pr[Y=1 \mid X=1]$$

g) $X \leftarrow \text{uniform}(1,2,\dots,n)$ $Y \leftarrow \text{uniform}(X, X+1, \dots, n)$

$$f_X(x,y) = \Pr[X=x \wedge Y=y] = \Pr[Y=y \mid X=x] \cdot \Pr[X=x]$$

$$= \frac{1}{n-x+1} \cdot \frac{1}{n}$$

$$= \frac{1}{n^2 - nx + n}$$

$$\Pr[\text{2 Elemente auf gleiche hasht}] = \sum_{h=1}^{n} \frac{1}{n} \cdot \frac{1}{n} = \frac{1}{n}$$

$$\Pr[\text{Irgend ein Paar } i<j \text{ hasht}] = \bigcup_{i=1} \leq \sum_{1 \leq i < j \leq n} \frac{1}{n} = \binom{k}{2} \cdot \frac{1}{n}$$

$$X(s_1, \dots, s_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} s_i s_j (v_i \cdot v_j)$$

$$\mathbb{E}[X(s_1, \dots, s_n)] = \sum \sum \mathbb{E}[s_i s_j (v_i \cdot v_j)]$$

Falls $i=j$ $\mathbb{E}[s_i s_j (v_i \cdot v_j)] = v_i^2$

Falls $i \neq j$ $= \frac{1}{2} \cdot 1 \cdot v_i v_j + \frac{1}{2}(-1) v_i v_j$

$$\mathbb{E}[X(\dots)] = \sum_{i=0}^{n} v_i^2$$

$\Pr[A] = \frac{1}{100}$ A = Bester unter 100 zufällig

$$\Pr[B] = \frac{1}{2000}$$

$$\Pr[B \mid A] = \frac{\Pr[A \mid B] \cdot \Pr[B]}{\Pr[A]} = \frac{1 \cdot \frac{1}{2000}}{\frac{1}{100}} = \frac{100}{2000}$$

Proof: Euler tour exists: As soon as one takes an edge from start only one edge with odd degree exists. Removing an edge will lead to the node dropping 2 degrees. Can only end when revisiting odd node.
$\Rightarrow$ Only even degrees, add subcycle from here.

Maximum spanning tree := Multiply edges with (-1)

Anzahl Hamiltonkreise (max.) $= \frac{(n-1)!}{2}$
$a|b\,d\,c|a = a|c\,d\,b|a$

Q_d indem: $000 \to 001 \to 011 \to 010 \to 110 \xrightarrow{\text{rückwärts}} 111 \to \cdots \to 100$

Widerlegargumente: Gittergraph Schachbrettargument.

Bipartit färben $\Rightarrow \#Rot \overset{!}{=} \#Blau + \{-1, 0, 1\}$ da Pfad abw.
Bzw. Hamiltonkreis $\#Rot = \#Blau$

Dirac: $|V| \geq 3 \wedge \underbrace{\min(\deg(v))}_{\delta(v) =} \geq \frac{|V|}{2} \Rightarrow$ Hamiltonisch
Beweis: x, y haben mind. Pfad von Länge 2, da $N(x) \cap N(y) \neq \emptyset$

x –···· i – $i+1$ –···– y

$N(x) \cup N(y)$ müssen in der Liste sein (wenn max)
Entweder x, y sind verbunden oder $k \geq \frac{|V|}{2} + 2$
Es existiert immer ein adjazentes Paar $i, i+1$ sodass $i - y \wedge i+1 - x$
Beweis: Markiere immer den linken Knoten von $N(x)$ weg. $\frac{|V|}{2} - \underset{\uparrow v_2, v_1}{1}$ Knoten.
Da aber $\frac{|V|}{2} + \frac{|V|}{2} - 1 + 2 > |V|$ ↯ $\square$ Also existiert, also

erst finden wir $v_1, \dots, v_k$ und dann ein v_i mit $v_i - y$ und $x - v_{i+1}$
Den können wir mit Kreis vergrössern. $O(|V|^2)$

$n^2 2^n$ Algorithmus mit $n 2^n$ Speicher:

$$P_{S,x} = \begin{cases} 1, & \exists \text{ Pfad von } 1 - x \text{ mit Knoten aus } S \\ 0, & \text{else} \end{cases}$$

$P_{S,x} = 1 :\Leftrightarrow 1 \in S \wedge x \in S \wedge \exists c \in N(x): P_{S \setminus \{x\}, c} = 1$
Initialisierung: $P_{\{1,x\},x} = 1$ (alle anderen 2-Tupel sind 0)
Alle Mengen rekursiv oder mit BFS generieren.

Hamilton Seq.	Lexikographisch sortierte Teilmengen:		S alle Mengen
0 0 0	0 0 0	$\emptyset$	0 0 0
1 0 0	0 0 1	a	0 0 1
1 1 0	0 1 1	ab	0 1 0
0 1 0	1 1 1	abc	1 0 0
0 1 1	1 0 1	ac	0 1 1
1 1 1	0 1 0	b	1 0 1
1 0 1	1 1 0	bc	1 1 0
0 0 1	1 0 0	c	1 1 1

Kein Algo.
BFS auf Q_d: 000 ; Kard. 1: 001, 010, 100 ; Kard. 2: 011, 101, 110 ; Kard. 3: 111

k-Zusammenhängend $\Rightarrow |X| < k \Rightarrow G[V \setminus X]$ zusammenhängend.
$\wedge\ |V| \geq k+1$ (mind. k Knoten).
2-zus $\Leftrightarrow$ 3-zus

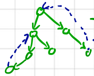
k-kanten-zusammenhängend $\Rightarrow X \subseteq E \wedge |X| < k \Rightarrow (V, E \setminus X)$ zus.hä.
$\wedge\ |E| \geq k+1$

Satz Menger

k zusammenhängend $\Leftrightarrow \forall u \neq v: |\text{intern-knotendisjunkte Pfade}| \geq k$

k kantenzusammenhängend $\Leftrightarrow \forall u \neq v: |\text{Kantendisjunkte Pfade}| \geq k$

Artikulationspunkt



→ Vorwärtskanten
⇢ Rückwärtskanten

$low[v]$:= Tiefste DFS-Nummerierung die man mit ∞-vielen Vorwärtskanten und einer Rückwärtskante findet.
Offensichtlich $\forall v: low[v] \leq dfs[v]$

Wurzel wird mit 1 markiert in Vorlesung.

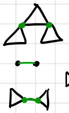
$\exists w \in \text{unvisited } N(v): low[w] \geq dfs[v] \Leftrightarrow v$ Artikulationsknoten (ausser Wurzel)

$$low[v] := \min\left(dfs[v],\ \min_{w \in N(v) \setminus \{pred(v)\}}(w[w])\right)$$

Wurzel Artikulationsknoten $\Leftrightarrow$ Wurzel hat mehr als 1 Kind im DFS-Baum.

$O(|V| + |E|)$, $O(|E|)$ falls zusammenhängend (duh)

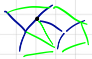
Eine Vorwärtskante ist genau dann eine Brücke wenn $dfs[v] > low[w]$



Eine Brücke kann nur zwischen Blätter und Artikulationsknoten liegen, dass ist aber nicht hinreichend.

$\delta(V)$ = Minimalgrad $\Delta(V)$ = Maximalgrad.

Maker-Breaker game:



Graph lässt sich in 2 Spannbäume teilen. Breaker beginnt. Maker baut Spannbaum wieder auf.

Non-metric TSP (Traveling Salesperson problem)
on a complete graph with:
If you can solve TSP you can solve Hamilton.

Proof: $\ell(a,b) := \begin{cases} 0 & \text{if a path from a to b exists} \\ 1 & \text{else} \end{cases}$

$\Rightarrow$ Kein α-Approximationsalgorithmus existiert.

Ein α-Approx. Algo würde ein C finden, sodass:

$\sum_{e \in C} \ell(e) \leq \alpha \cdot opt(K_n, \ell)$. Dies würde aber Hamiltonkreis Problem lösen, da $opt(K_n, \ell) = 0$!

Metric TSP auf kompletten Graphen:
$\forall u, v, w \in V: \ell(u,w) \leq \ell(u,v) + \ell(v,w)$

2-Approximation: MST aussenrum laufen. Knoten mehrmals besuchen überspringen.

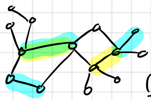
Matching: $M \subseteq E \wedge$ alle Kanten aus M teilen keinen inzidenten Knoten.
Perfektes Matching: Jeder Knoten wird von M überdeckt $\wedge |M| = |V|/2$
M ist **kardinalitätsmaximales** Matching, falls $\forall M': |M| \geq |M'|$
en: maximum Matching
M ist **inklusionsmaximales** Matching, falls $\forall e \in E \setminus \{M\}: M \cup \{e\}$ kein Matching ist.

Greedy-Matching

$M \leftarrow \emptyset$, while $(E \neq \emptyset)$ $\{e = (u,v) \in E,\ M \leftarrow M \cup \{e\},\ G \leftarrow G[V \setminus \{u,v\}]\}$

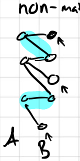
$2|M_{Greedy}| \geq |M_{max}| \geq |M_{Greedy}|$ (jede Kante aus M_{Max} berührt mind. eine Kante in M_{Greedy} ($|M_{max}| \geq |M_{Greedy}|$
jeder Knoten aus M_{Greedy} berührt eine Kante aus M_{Max} ($2|M_{Greedy}| \geq |M_{max}|$

$M_1 \cup M_2$ zwei Matchings $\rightarrow G_M = (V, M_1 \cup M_2)$



$\forall v \in G_M: \deg(v) \leq 2$
und G_M besteht nur aus (knotendisjunkten) Pfaden und Zyklen von gerader Länge.

Falls $|M_1| < |M_2|$, existiert ein Pfad ungerader Länge in $M_1 \cup M_2$
$M_1' := (M_2 \cup (P \cap M_2)) \setminus (P \cap M_1)$



Bipartites Matching $O(|V| \cdot |E|)$

BFS

Füge nicht überdeckte Knoten aus Menge A in die Queue. Wenn nach BFS Suche ein nicht überdeckter Knoten in B gefunden wurde, hat man ein augmentierendes Matching (Historie wird gespeichert).

Allgemein: $O(\sqrt{|V|}\,|E|)$ und $O(|V||E|\log|V|)$ gewichtet.

Christofides Algorithm $3/2$ opt (K_n)

1. $T = MST(G)$ $\quad O(E \log|V|)$ oder $O(V \log V + E)$
2. $V' := \{v : deg_T(v) \% 2 = 1\}$
3. M = Minimum perfect matching $[G[V']]$ $\quad O(\sqrt{|V|}\,E)$
4. $G^* := T \cup M$
5. Euler tour W auf G^* $\quad O(E)$
6. Kürzester Pfad durch W (Dijkstra shortcuts) $O(V \log V + E)$ $= C$

$\ell(T) := mst(G) \leq mhk(G)$

$\ell(M) \leq \frac{1}{2}\, mhk(G)$

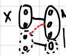
$\ell(W) = \ell(T) + \ell(M)$

$\ell(C) \leq \ell(W) \leq \frac{3}{2}\, mhk(G)$

- Satz von Hall / Heiratssatz

Für ein bipartiten Graph existiert ein Matching mit $|M| = |A|$ → M zusammenhängend

$\Leftrightarrow \forall X \subseteq A : |N(X)| \geq |X|$ $\quad$ Idee:

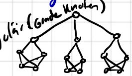


Also muss, da $|A \setminus X| \leq |N(A \setminus X)|$

- Ein k-regulärer bipartiter Graph kann in k-perfekte Matchings aufgeteilt werden. (Es gilt $\forall X : |N(X)| \geq |X| \Rightarrow$ perfektes Matching entf. $k-1$-regulärer bip. Graph

Ungerade Kreise und komplette Graphen mit ungeradem Grad haben kein perfektes Matching.

Gegenbsp. für 3-regulär (Grade Knoten)



Für bipartite Graphen ist das Min-Vertex Cover Problem identisch zum maximalen Matching Problem. Da jede Kante des Matching in einem Cover liegen muss gilt: $|\text{min cover}| \geq |\text{matching}|$

- Ist $G = (V, E)$ ein 2^k-regulärer bipartiter Graph so kann man in $O(|E|)$ ein Matching finden (genau in $2|E| - 1$)

Einfach Eulertour finden und jede zweite Kante löschen.

$2^k \to 2^{k-1} \to \dots \to 2^0 = 1$-regulär!

Färbung

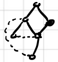
$\delta(G) :=$ Minimalgrad von G

$\Delta(G) :=$ Maximalgrad von G $\quad$ n partit $\Rightarrow n+1$-partit

$c : V \to [k]$ ist eine Färbung

$\chi(G)$ = Chromatische Zahl = Minimale # Farben.

- Ein Graph bipartit $\Leftrightarrow$ Keine Kreise ungerader Länge
  - $\Rightarrow$ Folgt direkt, da ungerade Kreise $\chi(G) = 3$ besitzen.
  - $\Leftarrow$ BFS von einem Knoten aus, Färben mit Distanz von s entfernt. Ohne Kreise ungerader Länge können diese Stufen nie zusammen kommen.



- $\chi(G) \leq k \Leftrightarrow G$ k-partit

- Jeder planare Graph ist 4-färbbar.
- Nur $K_{1,n}$ und $K_{2,n}$ sind planare bipartite komplette Graphen.

- Gegeben ein Graph, ist $\chi(G) \leq 3$ ist NP vollständig.
- Die Frage $\chi(G) < 3$ ist nicht NP-vollständig

Greedy Färbung:

Wähl eine beliebige Folge der Knoten und wähle die kleinste Farbe aus, die die Nachbaren nicht haben.

Färbt worst-case: $\Delta(G) + 1$

Kleinste Farbe die Nachbaren nicht haben $\Rightarrow$ HashMap rein und wieder löschen. Oder Array am Anfang auch mit wieder löschen!

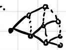
Es gibt eine Reihenfolge des Greedy-Algorithmus der $n!$ Farben verwendet. Use Knuth-Shuffle for this: Beweis: Erst alle 0, dann alle 1, usw.

`for(int i=0; i<=n; ++i){r=randInt(0,i); swap(arr[i],arr[j]);}`

Gegenbeispiel Greedy:

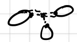
Kompletter bipartiter Graph ohne gegenüberliegende Kante mit $|A|$ Farben, obwohl 2 genügend wären. $\Delta(G) = 3$ färbt in 4.

- $\exists$ ein Knoten mit $deg(v) \leq \Delta(G) \Rightarrow$ DFS/BFS von v und dann Greedy-Algorithmus in umgekehrter Knotenrichtung.



Jeder Knoten v_i hat dann ein Nachbar v_j mit $j > i$. Der letzte Knoten hat $deg(v_n) < \Delta(G)$, also ✓

- Hat es ein Artikulationsknoten können wir G aufteilen.



Diese Teilgraphen mit v haben die Eigenschaft, dass $deg(v) < \Delta(G)$ und dessen Farben können permutiert werden, sodass v überall die gleiche Farbe hat. □

Es gibt k-reguläre Graphen für die $\Delta(G) + 1$ Farben notwendig sind (vollständige Graphen/ungerade Kreise).

Brooks hat bewiesen, dass das die einzigen Typen sind und dass sonst gilt: $\chi(G) \leq \Delta(G)$

Wenn in jedem induzierten Subgraph $\exists v : deg(v) \leq k \Rightarrow$ Eine $\Delta(G)$ Färbung finden. (Indem der Knoten entfernt wird (v_n) und die Grade neu gerichtet werden, etc. $(v_{n-1}, v_{n-2}, \dots)$.

Jeden 3 färbbaren Graphen kann man in $O(|E|)$ mit $\sqrt{|V|}$ Farben färben.

$$P(B) = P(A \cap B) + P(\bar{A} \cap B) = P(B|A) \cdot P(A) + P(B|\bar{A}) \cdot P(\bar{A})$$

Inverse:

$$P(B) = \frac{P(A) - P(A|\bar{B})}{P(A|B) - P(A|\bar{B})}$$

X - Anzahl ...

Falls $n = 2^k$

Probability that n is in middle 3

$\frac{1}{n}$

$\binom{n}{2} (1-p)^{n-i} (p)$

Ω = Ereignismenge = $\{\omega_1, \omega_2, \dots, \omega_n\}$, wobei ω_i Elementarereignis

$Pr: \Omega \to [0,1] \quad \sum_{\omega \in \Omega} Pr[\omega] = 1 \; ; \; Pr[E] = \sum_{\omega \in E} Pr[\omega]$

$\bar{E} := \Omega \setminus E \; ; \; Pr[\bar{E}] = 1 - Pr[E] \; ; \; Pr[\emptyset] = 0;$

Laplace := $\forall \omega \in \Omega: Pr[\omega] = \frac{1}{|\Omega|} \Rightarrow Pr[E] = \frac{|E|}{|\Omega|}$

Additionssatz: $Pr\left[\bigcup_{i=1}^{n / \infty} A_i\right] = \sum_{i=1}^{n / \infty} Pr[A_i]$! falls disjunkt

Siebformel: $Pr\left[\bigcup_{i=1}^{n} A_i\right] = \sum_{l=1}^{n} (-1)^{l+1} \cdot \left[\sum_{1 \le i_1 < \dots < i_l \le n} Pr[A_{i_1} \cap \dots \cap A_{i_l}]\right]$

(Prinzip der Inklusion-Exklusion)

$= \sum_{i=1}^{n} Pr[A_i] - \sum_{1 \le i_1 < i_2 \le n} Pr[A_{i_1} \cap A_{i_2}] + \dots$

Konkret: $Pr[A_1 \cup A_2] = Pr[A_1] + Pr[A_2] - Pr[A_1 \cap A_2]$

$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|$

$Pr[A \cap B \cap C] = Pr[A] + Pr[B] + Pr[C] - Pr[A \cap B] - Pr[A \cap C] - Pr[B \cap C] + Pr[A+B+C]$

Kombinatorik:

Variation mit Wiederholung n^k

" ohne " $\frac{n!}{(n-k)!}$

Kombination mit Wiederholung $\binom{n+k-1}{k} = \frac{(n+k-1)!}{(n-1)!\,k!}$

" ohne " $\binom{n}{k} = \frac{n!}{(n-k)!\,k!}$

Bedingte Wahrscheinlichkeiten:

$P_B(A) = P(A|B)$

$P(\bar{B}|A) = 1 - P(B|A)$

Satz der totalen Wahrscheinlichkeit:

$B \subseteq \biguplus_{i=1}^{n} A_i \Rightarrow Pr[B] = \sum_{i}^{n / \infty} Pr[B|A_i] = \sum_i P(A_i) \cdot P(B|A_i)$

Korrolar: $Pr[B] = Pr[B|A] \cdot Pr[A] + Pr[B|\bar{A}] \cdot Pr[\bar{A}]$

Satz von Bayes: $B \subseteq \biguplus_{i}^{n / \infty} A_i$

$\forall i: Pr(A_i|B) = \frac{Pr(A_i \cap B)}{Pr(B)} = \frac{Pr(B|A_i)\,Pr(A_i)}{\sum_{j=1}^{n / \infty} Pr(B|A_j)\,Pr(A_j)}$ $= P(A)$

$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{Pr(B|A) \cdot Pr(A)}{Pr(B)}$

$Pr(A_2|B) = \frac{Pr(B|A_2) \cdot P(A_2)}{Pr(B|A_1) \cdot P(A_1) + Pr(B|A_2)\,P(A_2) + P(B|A_3) \cdot P(A_3)}$

$P(A \cap B) = P(B|A) \cdot P(A) = P(A) + P(B) - P(A \cup B)$

Multiplikationssatz:

$P[A_1 \cap \dots \cap A_n] = P[A_1] \cdot P[A_2|A_1] \cdot P[A_3|A_1 \cap A_2] \cdots$

Beweis mittels kürzen:

$\frac{Pr[A_1]}{1} \cdot \frac{P[A_1 \cap A_2]}{P[A_1]} \cdot \frac{Pr[A_1 \cap A_2 \cap A_3]}{P[A_1 \cap A_2]} \cdot \dots \cdot \frac{Pr[A_1 \cap \dots \cap A_n]}{Pr[A_1 \cap \dots \cap A_{n-1}]}$

Unabhängigkeit: $A_1 \dots A_n$ falls:

Bsp.: $A = \emptyset, B = \{1,2\}, C = \{1,3\}$

Bsp.: $A = \{1,2,3\}$ $B = \{1,2,4\}, C = \{1,2,5\}$

Für jede endliche Teilmenge I gilt: $P[A_{i_1} \cap \dots \cap A_{i_k}] = P[A_{i_1}] \cdots P[A_{i_k}]$

Wenn A, B, C unabhängig sind $\Rightarrow A \cap B, C$ bzw. $A \cup B, C$ unabhängig.

Birthday problem: m Bälle $\le n$ Körbe

Wahrscheinlichkeit das nach m-Mal werfen, in einem Korb 2 Bälle auftreffen. (Für $m \ge n$ immer).

$A_i := 1$, falls Ball i landet in leerem Korb.

$Pr\left[A_j \,\middle|\, \bigcap_{i=1}^{j-1} A_i\right]$ = W'keit das j-ter Ball in leere Urne landet.

$= \frac{n-(j-1)}{n} = 1 - \frac{j-1}{n}$

$Pr[A] = Pr\left[\bigcap_{i=1}^{m} A_i\right] = Pr[A_1] \cdot Pr[A_2|A_1] \cdots Pr\left[A_m \,\middle|\, \bigcap_{i=1}^{m-1} A_i\right]$

$\prod_{j=1}^{m} \left(1 - \frac{j-1}{n}\right) \le \prod_{j=2}^{m} e^{-\frac{j-1}{n}} = e^{-\frac{1}{n} \sum_{j=1}^{m-1} j} = e^{-\frac{m(m-1)}{2n}}$

$1 - x \le e^{-x}$

Wie oft bis Muster erreicht? X_i = Schritte bis i-ter Teil von Muster erreicht.

$\mathbb{E}[X_i] = \frac{1}{2}(\mathbb{E}[X_{i-1}] + 1) + \frac{1}{2}(\mathbb{E}[X_{i-1}] + 1 + \mathbb{E}[X_i])$

Zufallsvariablen

$X: \Omega \to \mathbb{R}$ Zufallsvariable (muss auf ganz Ω def. sein!)

Indikatorvariable $X_A(\omega) = \begin{cases} 1, & \text{falls } \omega \in A \\ 0, & \text{sonst} \end{cases}$ $\quad \mathbb{E}[X_A] = P[A]$

$W_X := X(\Omega)$ = Wertebereich

$X^{-1}: \mathbb{R} \to \Omega$ (Umkehrabbildung)

$Pr[\text{"}X = x\text{"}] = Pr[X^{-1}(x)]$ wobei $x \in W_X$

$Pr[\text{"}X \le x_i\text{"}] = \sum_{x \in W_X: x \le x_i} Pr[\text{"}X = x\text{"}] = 1 - \sum_{x \in W_X: x > x_i} Pr[\text{"}X = x\text{"}]$

Dichtefunktion / Randdichte (PDF)

$f_X: \mathbb{R} \to [0,1]; \; x \mapsto Pr[X = x]$ bzw. $f_X(\omega) = Pr[\text{"}X = \omega\text{"}]$

z.B. $\Omega = \{⚀, \dots, ⚅\}$

$X(\omega) = \begin{cases} 1, & \text{falls } \omega \text{ prime ungerade Augenzahl hat} \\ 0, & \text{sonst} \end{cases}$

$f_X(1) = \frac{|\{⚂, ⚄\}|}{6} = \frac{1}{3} \; ; \; f_X(0.5) = 0$

$f_X(0) = \frac{2}{3}$

Verteilungsfunktion (CDF)

$F_X: \mathbb{R} \to [0,1]; \; x \mapsto Pr[X \le x] = \sum_{x' \in W_X: x' \le x} Pr[X = x']$

$F_X(1.5) = 1$, da $f_X(0) = \frac{2}{3}$ und $f_X(1) = \frac{1}{3}$

Zugrundeliegender Wahrscheinlichkeitsraum:

$W_X = \{0, 1\}$

$Pr[X = 0] = \frac{2}{3} = f_X(0)$

Erwartungswert

$\mathbb{E}[X] := \sum_{x \in W_X} x \cdot Pr[\text{"}X = x\text{"}]$ = Bsp.: $0 \cdot \frac{2}{3} + 1 \cdot \frac{1}{3} = \frac{1}{3}$

Bsp. nicht konvergierenden unendlichen Erwartungswert

Bankspiel Zufallsvariable Gewinn:

$G := \begin{cases} 2^k, & \text{falls } k \text{ ungerade} \\ 2^{-k}, & \text{falls } k \text{ gerade} \end{cases}$

$Pr[\text{"Anzahl Würfe} = k\text{"}] = \left(\frac{1}{2}\right)^k$, also $\underbrace{Z\,Z \dots Z}_{k-1 \text{ mal}}\,K$

$\sum_{k=1}^{\infty} (-1)^{k-1} \cdot 2^k \left(\frac{1}{2}\right)^k = 1 - 1 + 1 - 1 \dots$

Boolesche Ungleichung:

$Pr\left[\bigcup_{i=1}^{n} A_i\right] \le \sum_{i=1}^{n} Pr[A_i]$

$Pr\left[\bigcap_{i=1}^{n} A_i\right] \le \min Pr[A_i]$

Standard:

$1 - x \le e^{-x} \quad \forall x \in \mathbb{R}$

$\left(\frac{e \cdot n}{k}\right)^k \quad \forall n, k \in \mathbb{N}$

Erwartungswert: $\boxed{W_X \leq \mathbb{N}_0}$

$$\mathbb{E}[X] = \sum_{i=1}^{\infty} Pr[X \geq i] = \sum_{i=0/1}^{\infty} i \cdot Pr[X=i]$$

Beweis: $\sum_{i=0}^{\infty} i \cdot Pr[X=i] = \sum_{i=0}^{\infty} \sum_{j=1}^{i} Pr[X=i] =$ (1,1) (2,1) (2,2) (3,1) ⋮

$$= \sum_{j=1}^{\infty} \sum_{i=j}^{\infty} Pr[X=i] = \sum_{j=1}^{\infty} Pr[X \geq j]$$

Wichtig $Pr[A \geq B] = 1 - Pr[A < B]$

Markov-Ungleichung $X \geq 0, t > 0$

$$Pr[X \geq t] \leq \mathbb{E}[X]/t$$

Proof: $\mathbb{E}[X] = \sum_{i=1}^{\infty} Pr[X \geq i] \geq \sum_{i=1}^{t} Pr[X \geq i] \geq t \cdot Pr[X \geq i]$

Linearität vom Erwartungswert!

Gegeben n Zufallszahlen: $X_i : \Omega \to \mathbb{R}$

$$X = \alpha_1 X_1 + \alpha_2 X_2 + \dots + \alpha_n X_n + b$$

$$\mathbb{E}[X] = \alpha_1 \mathbb{E}[X_1] + \dots + \alpha_n \mathbb{E}[X_n] + b$$

Bsp.: Sehfahrer

X = Jeder Sehmann ist in seinem eigenen Bett.

$$X_i := \begin{cases} 1, & i\text{-te Seemann ist in seinem Bett} \\ 0, & \text{sonst} \end{cases}$$

$$X = X_1 + X_2 + \dots + X_n$$

$$\mathbb{E}[X_i] = 0 \cdot Pr[X_i = 0] + 1 \cdot Pr[X_i = 1] = \frac{1}{n}$$

$\mathbb{E}[X] = n \cdot 1/n = 1$ (Durchschnittlich ein Seemann).

Algorithmus Stabile Menge (möglichst grosse Menge, sodass $G[S]$ keine Kanten hat)

1. Lösche Knoten v mit Wahrsch. p
2. Jede Kante im übriggebliebenen Graphen löscht einen Nachbarsknoten.

V = Anzahl Knoten die übrig sind nach Alg.

$$Y_v = \begin{cases} 1, & v \text{ löscht sich} \\ 0, & \text{sonst} \end{cases} \qquad \mathbb{E}[Y_v] = 1 \cdot p + 0 \cdot (1-p) = p$$

$$Z_e = \begin{cases} 1, & \text{Kante } e \text{ überlebt erste Runde} \\ 0, & \text{else} \end{cases} \qquad \mathbb{E}[Z_e] = (1-p)^2$$

$V \geq n - \sum_v Y_v - \sum_e Z_e \Rightarrow$ jede Kante löscht Knoten.

= falls zwei Kanten nicht den gleichen löschen würden.

$\mathbb{E}[V] \geq n - n \cdot p - m \cdot (1-p)^2$ $\quad \frac{n \cdot d}{2}$ Kanten

$S \geq Q - R$ Q = Wahrsch. das Knoten überlebt.

R = Wahrsch. das Kante Knoten löscht. $\mathbb{E}[S] \geq np - mp^2$

Ableiten $2mp - n \overset{!}{=} 0 \Rightarrow p = \frac{n}{2m}$ optimal

Falls Graph d-regulär ($\frac{n \cdot d}{2}$ Kanten): $p = \frac{n}{2 \frac{n \cdot d}{2}} = \frac{1}{d}$

Falls $p = \frac{n}{2m} \Rightarrow \mathbb{E}[|X|] = \frac{n^2}{4m}$

$Pr[\,|S| \geq \frac{n^2}{4m}] \geq 1 - 10^{-50}$

$Pr[$ Alg. findet keine Menge der Gr. $(1-\varepsilon) \cdot \frac{n^2}{4m}] \leq f(\varepsilon)$

$Pr[$ Alg. " keine " .. $(1-\varepsilon) \frac{n^2}{4m}$ bei N aufrufen$] \leq (f(\varepsilon))^n$

2.36 fehlt

Durchsch. Distanz von $\mathbb{E}[X]$ entfernt quadriert.

$$Var[X] := \mathbb{E}[(X - \mathbb{E}[X])^2] = \sum_{x \in W_X} (x - \mu)^2 \cdot Pr[X=x]$$

$\mu = \mathbb{E}[X]$

$\sigma = \sqrt{|Var[X]|}$ = Standardabweichung ≈ Durchsch. Distanz von $\mathbb{E}[X]$

$$Var[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2 \quad !!!!!$$

$$Var[\alpha X + B] = \alpha^2 \, Var[X] \qquad \sigma(\alpha X + b) = |\alpha| \, \sigma(X)$$

$\mathbb{E}[X^k]$ = k-te Moment, $\mathbb{E}[X]$ = 1st zentraler Moment.

$\mathbb{E}[(X - \mathbb{E}[X])^k]$ = k-te zentrale Moment

$$Var[2X] = \mathbb{E}[(2X - 2\mathbb{E}[X])^2] = \mathbb{E}[4(X - \mathbb{E}[X])^2] = 4\,Var[X]$$

$$X \sim Bernoulli(X) :\Leftrightarrow f_X(x) = \begin{cases} p, & \text{wenn } X=1 \\ 1-p, & \text{wenn } X=0 \\ 0, & \text{sonst} \end{cases}$$

Z.B.: Indikatorvariable von einem Münzwurf $\quad F[X \leq 1] = 1$, $F[X \leq 0.5] = 0$

$\mathbb{E}[X] = p$ und $Var[X] = p(1-p)$ und $\mathbb{E}[X^2] = p$

N-mal Münzwurf, $X = \#$ Kopf, $W_Y = \{0, 1, .., n\}$

$$X \sim Bin(n,p) :\Leftrightarrow f_X(x) = \begin{cases} \binom{n}{x} p^x (1-p)^{n-x}, & \text{für } x \in W_X \\ 0 & \text{sonst} \end{cases}$$

$\mathbb{E}[X] = np$, $Var[X] = np(1-p)$

$$F_X(x) = P(X \leq x) = \sum_{k=0}^{\lfloor x \rfloor} \binom{n}{k} p^k (1-p)^{n-k}$$

Geburtstagproblem: M-Bälle in n-Körbe

X = Wahrsch. erster Ball im 1. Korb
$X \sim Bernoulli(\frac{1}{n})$

Y = # Bälle im ersten Korb
$Y \sim Bin(m, \frac{1}{n})$

Z = # Körbe ohne Ball

Z_i = i-ter Korb leer
$Z_i \sim Bernoulli((\frac{n-1}{n})^m)$

$Z = \sum Z_i$; $\mathbb{E}[Z_i] = \sum \mathbb{E}[Z_i]$

X = Anzahl Versuche bis gelungenes Resultat

$$X \sim Geo(p) \Leftrightarrow f_X(i) = \begin{cases} \underbrace{p(1-p)^{i-1}}_{= Pr[X=i]}, & \text{für } i \in \mathbb{N} \\ 0, & \text{sonst} \end{cases}$$

$\mathbb{E}[X] = \frac{1}{p}$; $Var[X] = \frac{1-p}{p^2}$

$$F_X(n) = P[X \leq n] = \sum_{i=1}^{n} P[X=i] = 1 - (1-p)^n$$

$$P[X \geq t] = (1-p)^{t-1}$$

Gedächtnislosigkeit ~ X

$$P[X \ge s+t \mid X > s] = P[X \ge t]$$

$Geo(p) \sim X \Rightarrow$ Gedächtnislosigkeit X

Beweis: $P[X \ge s+t \mid X > s] = \frac{P[X \ge s+t]}{P[X>s]}$

$$= \frac{(1-p)^{s+t-1}}{(1-p)^{s}} = (1-p)^{t-1} = Pr[X \ge t]$$

Geom allgemein:

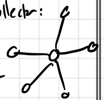
V_n = Anzahl Würfe bis zum n-ten Erfolg

$E[V_n] = \frac{n}{p}$ ⇝ Aus Linearität: $\underbrace{E[Z]+\dots+E[Z]}_{n\text{-mal}}$ geom. Verteilung

$$f_Z(z) = \binom{z-1}{n-1} p^n (1-p)^{z-n}$$

Coupon-Collector

Coupon Collector:



X = Anzahl Käufe bis zur Komplettierung

$$\underbrace{2}_{1},\ \underbrace{2,1}_{2},\ \underbrace{2,2,3}_{3},\ \underbrace{1,3,2,3,1,4}_{4}$$

X_i = Anzahl Käufe bis ein neuer Coupon kommt, gegeben $i-1$ Coupons:

$$X_i \sim Geo\left(\frac{n-(i-1)}{n} = \frac{n-i+1}{n}\right)$$

$$E[X_i] = \frac{1}{p} = \frac{n}{n-i+1}$$

$$X = \sum X_i \Rightarrow$$

$$E[X] = \sum E[X_i] = n \cdot \sum_{i=1}^{n} \frac{1}{n-i+1} = n \cdot \sum_{i=1}^{n} \frac{1}{i} \approx n \ln(n) + O(1)$$

Poisson-Verteilung

$$Po(\lambda) \sim X := f_X(i) = \begin{cases} \frac{e^{-\lambda} \cdot \lambda^i}{i!} & , i \in \mathbb{N}_0 \\ 0 & , \text{sonst} \end{cases}$$

$E[X] = Var[X] = \lambda$ $\quad E[X^2] = \lambda^2 + \lambda$

$Bin(n, \frac{\lambda}{n})$ konvergiert die Dichtefunktion nach $f_X(i)$!

Falls also n riesig, kann Approx. genutzt werden.

Mehrere Zufallsvariabeln

$\Omega = \{KKK, KKZ, KZK, ZKK, ZZK, ZKZ, KZZ, ZZZ\}$

X = Anzahl Zahl der ersten beiden Würfen

$W_X = \{0,1,2\}$

Y = Anzahl Zahl der letzten beiden Würfen.

$W_Y = \{0,1,2\}$

$Pr(X=0, Y=0) = \frac{|\{KKK\}|}{8} = 1/8$

$Pr(X=1, Y=0) = \frac{|\{ZKK\}|}{8} = 1/8$

$Pr(X=1, Y=1) = \frac{|\{ZKZ, KZK\}|}{8} = 1/4$

Gemeinsame Dichte von X und Y

$f_{X,Y}(x,y) = Pr[X=x, Y=y]$

$f_X(x) = \sum_{y \in W_Y} f_{X,Y}(x,y)$ (Randdichte berechnen)

$Pr(``X=x") = \sum_{y \in W_Y} Pr(``X=x, Y=y")$

Gemeinsame Verteilung

$F_{X,Y}(x,y) := Pr[X \le x, Y \le y] = \sum_{x' \in W_X, x' \le x} \sum_{y' \in W_Y, y' \le y} f_{X,Y}(x',y')$

$F_X(x) = \sum_{x' \le x} f_X(x') = \sum_{x' \le x} \sum_{y \in W_Y} f_{X,Y}(x',y)$

Unabhängigkeit von diskreten Zufallsvariabeln

$X_1, X_2, \dots X_n$ sind unabhängig $\Leftrightarrow \forall (x_1 \in W_{X_1}, \dots, x_n \in W_{X_n})$

$Pr[X_1 = x_1, X_2 = x_2, \dots, X_n = x_n] = Pr[X_1 = x_1] \cdot Pr[X_2 = x_2] \cdots Pr[X_n = x_n]$

$f_{X_1, \dots, X_n}(w_1, w_2, \dots, w_n) = f_{X_1}(w_1) \cdot f_{X_2}(w_2) \cdots f_{X_n}(w_n)$

⇒ 2.53 Jede Teilmenge ist ebenfalls unabhängig

2.52 falls: $X_1 \dots X_n$ unabh. $\wedge\ S_1 \subseteq W_{X_1} \dots S_n \subseteq W_{X_n}$

$Pr[X_1 \in S_1, \dots, X_n \in S_n] = \sum_{x_1 \in S_1} \cdots \sum_{x_n \in S_n} Pr[X_1 = x_1, \dots, X_n = x_n]$

$\overset{\text{unabh.}}{=} \sum_{x_1 \in S_1} \cdots \sum_{x_n \in S_n} Pr[X_1 = x_1] \cdots Pr[X_n = x_n]$

$\overset{\text{Spezial}}{=} \left(\sum_{x_1 \in S_1} Pr[X_1 = x_1]\right) \cdots \left(\sum_{x_n \in S_n} Pr[X_n = x_n]\right) = Pr[X_1 \in S_1] \cdots Pr[X_n \in S_n]$

2.54 $f_i: \mathbb{R} \to \mathbb{R}$, $X_1 \dots X_n$ unabhängig $\Rightarrow f(X_1) \dots f_n(X_n)$ unabh.

Beweis mit 2.52 $z_i \in W_{f(X_i)}$: $S_i = \{x \mid f(x) = z_i\}$

- X, Y unabhängig $\wedge\ X+Y=Z \Rightarrow f_Z(z) = \sum_{x \in W_X} f_X(x) \cdot f_Y(z-x)$

Bsp.: $X \sim Poi(\lambda)$ $Y \sim Poi(\lambda) \Rightarrow$ $Z = X+Y$ $\sum_{x \in W_X} \frac{e^{-\lambda} \lambda^x}{x!} \cdot \frac{e^{-\lambda} \lambda^{(z-x)}}{(z-x)!}$

$= \frac{e^{-\lambda-\lambda}}{z!} \sum \binom{z}{x} \lambda^x \lambda^{z-x} = \frac{e^{-\lambda-\lambda}}{z!} (\lambda+\lambda)^z \Rightarrow Z \sim Poi(\lambda+\lambda)$

↳ Gilt nicht für Konstanten $z = 1+X+Y \not\Rightarrow z \sim Poi(1+x+y)$

$E[a_1 X_1 + \dots + a_n X_n + b] = a_1 \cdot E[X_1] + \dots + a_n E[X_n] + b$

$X_1, \dots, X_n$ unabhängig $\Rightarrow E[X_1 \cdot X_2 \cdots X_n] = E[X_1] \cdots E[X_n]$

$X_1, \dots, X_n$ unabhängig $\Rightarrow Var[X_1 + \dots + X_n] = Var[X_1] + \dots + Var[X_n]$

NIE ⇒ Mult. von Varianz

$Var[X-Y] = Var[X] + Var[-Y] = Var[X] + (-1)^2 Var[Y] = Var[X] + Var[Y]$

Waldsche Identität

N, X unabhängig $\wedge\ W_N \subseteq \mathbb{N}\ \wedge\ Z := \sum_{i=1}^{N} X_i$ ⇝ $X_i = X$ unabh. i.i.d. (indep. and identically distributed)

$\Rightarrow E[Z] = E[N] \cdot E[X]$

N = Anzahl Würfe bis Kopf, Z = Anzahl Kopf in weiteren N Würfe

$\sum_{i=1}^{N} X_i =: Z \Rightarrow E[X] = 1/2$ $\quad E[N] = 2;$ $\quad E[Z] = 1$

Falls X und Y unabhängig $\Rightarrow E[X|Y] = E[X]$

$E[X \mid Z=z] = \sum_{x \in \Omega} E[X=x \mid Z=z] Pr(Z=z)$ $\quad | \quad E[E[X|Y]] = E[X]$

$E[\alpha X + \beta Y \mid Z=z] = \alpha E[X \mid Z=z] + \beta E[Y \mid Z=z]$

$A(I,R)$ :: Korrektheit hängt von Eingabe I $\wedge$ Zufallsvariable R ab.

$Pr_R[A(I,R)$ ist korrekt$] \geq 1-\varepsilon$

Laufzeit: $\mathbb{E}[\text{Laufzeit}] = f(|I|)$ oder

$Pr_R[\text{Laufzeit} \leq f(n)] \geq 1-\varepsilon$

Korrektheit $\geq 10^{-4}$

$Pr[\text{mind. ein korrektes Resultat}] \geq 1-\underset{\substack{\uparrow \\ Pr[\text{Nicht korrekt}]}}{(1-10^{-4})}^{N}$ N_{mal}

Markov Ungleichung

X Zufallsvariable sodass $\forall \omega \in \Omega$ $X(\omega) \geq 0 \wedge t > 0$

$Pr[X \geq t] \leq \frac{\mathbb{E}[X]}{t}$ äquivalent $t = \mathbb{E}[X]t'$

$Pr[X \geq t' \mathbb{E}[X]] \leq \frac{1}{t'}$

Proof: $\mathbb{E}[X] = \sum_{i=0}^{\infty} Pr[X \geq i] \geq \sum_{i=0}^{t} Pr[X \geq i] \geq t \cdot Pr[X \geq t]$
(für $W_X \subseteq \mathbb{N}$)

Chebysher-Ungleichung

Sei $t > 0$

$Pr[|X-\mathbb{E}[X]| \geq t] \leq \frac{Var[X]}{t^2}$

$= Pr[(X-\mathbb{E}[X] \geq t) \vee (\mathbb{E}[X]-X \geq t)]$

Äquivalent $Pr\left[|X-\mathbb{E}[X]| \geq t \cdot \sigma[X]\right] \leq \frac{1}{t^2}$

Korollar: $Pr[X \geq (1+\varepsilon)\mathbb{E}[X]] \leq \frac{Var[X]}{(\varepsilon \cdot \mathbb{E}[X])^2}$

Beweis: $Y = (X-\mathbb{E}[X])^2$ mit $\mathbb{E}[Y] = Var[X]$

$Pr[|X-\mathbb{E}[X]| \geq t] = Pr[Y \geq t^2]$

$\leq \frac{\mathbb{E}[Y]}{t^2} = \frac{Var[X]}{t^2}$

Chernoff

$X = \sum_{i=1}^{n} X_i$, wobei $X_i(\omega) \in \{0,1\} \wedge$ Unabhäng.
bzw. $X_i \sim$ Bernoulli

$\mathbb{E}[X] = \sum_{i=1}^{n} p_i$

$\forall \varepsilon \in (0,1]$

$Pr[X \geq (1+\varepsilon)\mathbb{E}[X]] \leq e^{-\mathbb{E}[X] \cdot \frac{\varepsilon^2}{3}}$ $\forall \varepsilon > 0$

$Pr[X \leq (1-\varepsilon)\mathbb{E}[X]] \leq e^{-\mathbb{E}[X] \cdot \frac{\varepsilon^2}{2}}$ $\forall \varepsilon \in (0,1)$

$Pr[|X-\mathbb{E}[X]| \geq \varepsilon\mathbb{E}[X]] \leq 2e^{-\mathbb{E}[X] \cdot \frac{\varepsilon^2}{3}}$ $\forall \varepsilon \in (0,1)$

$Pr[X \geq t] \leq 2^{-t}$ $\forall t \geq 2e\mathbb{E}[X]$

Las-Vegas (Bsp. Randomisierter Quicksort)
Keine Fehler, aber die Laufzeit ist nicht deterministisch

Ziel: $\mathbb{E}[\text{Laufzeit}] \stackrel{!}{=}$ winzig

$\Rightarrow$ Monte Carlo, falls zu lange ???

Monte-Carlo (Bsp. Primzahltest)
Existiert Laufzeit, aber macht Fehler.

Ziel: $Pr[\text{"Fehler"}] \stackrel{!}{=}$ winzig

```
Quicksort(A,l,r):
    p := A[uniform(l,r)];
    int li,ri;
    k= Partitioniere <p und >p
    Quicksort(A,l,k-1);
    Quicksort(A,k+1,l);
```

OBDA $T_{l,r} = T_{r-l+1}$

$T_n = \#$ Schlüsselvergleiche $\quad t_n = \mathbb{E}[T_n]$

$t_n = \sum_{k=1}^{n} \mathbb{E}[T_n \mid \text{Pivot el. ist das } k\text{-kleinste El.}] \cdot \underbrace{Pr[\ldots]}_{\frac{1}{n}}$

$\leq \frac{1}{n}\sum_{k=1}^{n} \mathbb{E}[\underset{\substack{\uparrow \\ \text{Partitioniervergleiche}}}{(n-1)} + T_{k-1} + T_{n-k}]$

$t_n = \frac{1}{n}\sum_{k=1}^{n}((n-1)+t_{k-1}+t_{n-k}) \approx 3/2\, n\log(n)$

$t_n = n-1+\frac{2}{n}\sum_{i=0}^{n-1} t_i = n-1+\frac{2}{n}(t_1+t_2+\ldots)$

$\sum_{i=0}^{n} t_i = n-1+\frac{2}{n}\sum_{i=0}^{n-1} t_i + \sum_{i=0}^{n-1} t_i$ $k(n) = n-1+k(n-1)$ $\sim n^2\log(n)$

Quickselect

Partitioniere mit Pivot und schaue in welcher Seite das Median vorkommt.

```
if (k<t) QS(l,t-1) else if(k>t) Qs(t+1,r)
else if(k==t) return t;
```

$T_{n,k} = \#$ Schlüsselvergleiche (für n,k)

50%

X_i = Indikator i-te Pivot in Mitte

$X_i \sim$ Bernoulli$(1/2)$

Falls $X_i = 1 \Rightarrow$ höchstens $\frac{3}{4} \cdot n$ Element zum überprüfen.

Phase k :: Rekursive Aufrufe zw. $(k-1)$-ten $^{+1}$ und k-ten X_i welches 1 war.

$\underbrace{01}_{1}\underbrace{001}_{2}\underbrace{1}_{3}$ $\quad Y_k$ = Länge dieser k-ten Phase

$\#$ Schlüsselvergleiche k-ter Phase $\leq Y_k \cdot (3/4)^{k-1} \cdot n$

$\mathbb{E}[T_{n,\hat{k}}] \leq n \cdot \sum_{k \geq 1}^{\ell} \underbrace{\mathbb{E}[Y_k]}_{2} (3/4)^{k-1} \xrightarrow{\ell \to \infty} 8n$

2.70

Las Vegas

Für $N = \lceil \frac{1}{\varepsilon}\ln(\frac{1}{\delta}) \rceil$ Aufrufe:

$Pr[A \text{ korrekt}] \geq \varepsilon \Rightarrow Pr[A_\delta \text{ korrekt}] \geq 1-\delta$

Beweis:

$Pr[A_\delta \text{ falsch}] \leq (Pr[A \text{ falsch}])^N \leq (1-\varepsilon)^N \leq e^{-\varepsilon N} \leq \delta$

Satz kleinen Fermat: n prim $\Rightarrow \forall a \in \{1,\ldots,n\}: a^{n-1} \equiv_n 1$

Miller-Rabin: $O(\log(n))$ mit Primzahl 100%, keine Primzahl $\geq 3/4$

Target Shooting $\quad$ Bsp: $|S| = \pi 1^2$, $|U| = 2 \cdot 2 \Rightarrow \mathbb{E}[S] = \frac{\pi}{4}$

$S_i = \begin{cases} 1 & u_i \in S \\ 0 & \text{sonst} \end{cases}$

ABER $p = |S|/|U|$ für $|S|=1$, $|U|=\infty$ $\quad$ Bernoulli(S_i)

$S = \frac{1}{N}\sum_{i=1}^{N} S_i \quad \mathbb{E}[S] = \frac{1}{N} \cdot N \cdot \frac{|S|}{|U|} = \frac{|S|}{|U|}$

$Var[S] = \frac{1}{N^2} N \cdot Var[S_i] = \frac{1}{N}\frac{|S|}{|U|}\left(1-\frac{|S|}{|U|}\right) = \frac{1}{N}\left(\frac{|S|}{|U|}-\frac{|S|^2}{|U|^2}\right)$

$Var[\frac{1}{n}\sum_{i=1}^{N} S_i] = \frac{1}{n^2}\sum p(1-p)$

Wähle $\delta, \varepsilon > 0$ Wähle $N \geq 3\boxed{\frac{|U|}{|S|}} \cdot \varepsilon^{-2} \cdot \log(2/\delta)$

Wahrsch. δ dass: $\mathbb{E}[S] = \left[(1-\varepsilon)\frac{|S|}{|U|}, (1+\varepsilon)\frac{|S|}{|U|}\right]$

Beweis Chernoff!

Primzahltest $\Theta(\log(n))$ mit Wahrsch. p

$Pr[\text{Falsche Antwort} \mid \text{Antwort} = \text{kein Primzahl}] = 0$

$Pr[\text{Falsche Antwort} \mid \text{Antwort} = \text{Primzahl}] \leq 1/4$

Einseitiger Fehler

$Pr[A(I) \text{ korrekt} \mid A(I) = JA] = 1$ $(Pr[\text{falsch} \mid A = JA] = 0)$

$Pr[A(I) \text{ korrekt} \mid A(I) = NEIN] \geq \varepsilon$ $(Pr[\text{falsch} \mid A = NEIN] \leq 1-\varepsilon)$

$N \geq \frac{1}{\varepsilon} \log(1/\delta)$, $\mathbb{E}[A] = \frac{1}{\varepsilon}$, $(1-\varepsilon)^N \leq (e^{-\varepsilon})^N = \delta$

$\Rightarrow Pr[A_\delta(I) \text{ korrekt} \mid A(I) = NEIN] \geq 1-\delta$ nach N Wdh.

Zweiseitiger Fehler

$Pr[\text{Antwort falsch}] \leq 1/2 - \varepsilon \Leftrightarrow Pr[\text{Antwort korrekt}] = 1/2 + \varepsilon$

$N \geq 4/\varepsilon^2 \log(1/\delta) \Rightarrow Pr[\text{Antwort falsch}] \leq \delta$

Randomisierte Algo. für Minimierungsprobleme

$Pr[\text{Antwort} < f(\ell)] \leq 1-\varepsilon \wedge N \geq \frac{1}{\varepsilon} \log(1/\delta)$

$\Rightarrow Pr[\text{Antwort} < f(\ell)] \leq \delta$

z.B. Stabile Menge

Balls & Bins

$X_i \sim Bin(n, \frac{1}{n})$

$X = \max_{1 \leq i \leq n} X_i$ $\quad X_i := \#$ Bälle im i-ten Korb

$Y_{i,t} \hat{=} \begin{cases} 1, & \text{falls } X_i \geq t \\ 0, & \text{sonst} \end{cases}$ $\quad Y_t = \sum_{i=1}^{n} Y_{i,t} = \#$ Körbe mit $\geq t$

$Y_t > 0 \Leftrightarrow \max_{1 \leq i \leq n} X_i \geq t$

$\mathbb{E}[Y_t] = \sum_{i=1}^{n} \mathbb{E}[Y_{i,t}] \overset{\text{sym}}{=} n \cdot \mathbb{E}[Y_{1,t}] = n \cdot Pr[X_i \geq t]$

$\binom{n}{t} (\frac{1}{n})^t \cdot 1 \geq Pr[X_i \geq t] \geq Pr[X_i = t] = \binom{n}{t} (\frac{1}{n})^t (1-\frac{1}{n})^{n-t} \geq ?$

Unterscheidet sich um $(1-\frac{1}{n})^n \to \frac{1}{e}$

Ausserdem $\binom{n}{t} \leq (\frac{e \cdot ?}{k})^k \quad \forall 1 \leq k \leq n$

$? \leq Pr[X_i \geq t] \leq \binom{n}{t} (\frac{1}{n})^t \leq (\frac{e \cdot ?}{t})^t (\frac{1}{n})^t$

$t^t = n \Rightarrow t = (1-\varepsilon) \frac{\ln(n)}{\ln(\ln(n))} \to +\infty$; $t = (1-\varepsilon) \frac{\ln(n)}{\ln(\ln(n))} \to 0$

Also $t^t \leq \frac{\ln(n)}{\ln(\ln(n))}$

Für $m = n$ Bälle:

$\mathbb{E}[\max_{1 \leq i \leq n} X_i] \approx \frac{\ln(n)}{\ln(\ln(n))}$ relativ schnell ↗

Wähle 2 zufällige Körbe aus, Frag 2 Körbe und wirf ihn in den Korb mit den wenigeren Bälle.

Markov

$Pr[\text{korrekt}] = 0.6$

Mehrheitsentscheid in k-Läufe

$X_i := \begin{cases} 1, & \text{Antwort im i-ten Lauf korrekt} \\ 0, & \text{" " " Lauf falsch} \end{cases}$

$\sum_{i=1}^{k} X_i \geq 1/2 \rightsquigarrow$ Geben wir korrekte Antwort aus

$\mathbb{E}[\sum_i X_i] = k \cdot \mathbb{E}[X_i] = 0.6 \cdot k$

Wir wollen: $Pr[\sum_i X_i < k/2] <$ klein

$Pr[\sum^k X_i < (1-\varepsilon)\mathbb{E}[X]] \leq e^{-1/2 \varepsilon^2 \mathbb{E}[X]}$

$\Rightarrow (1-\varepsilon) 0.6k = 1/2 k \Rightarrow \varepsilon = 1/6$

Randomisierte Algo. für Maximierprobleme

$Pr[\text{Antwort} \geq f(\ell)] \geq \varepsilon \wedge N \geq \frac{1}{\varepsilon} \ln(\frac{1}{\delta})$

$\Rightarrow Pr[\text{Antwort} \geq f(\ell)] \geq 1-\delta$

Satz von Fermat

$0 \leq a \leq n$

n prim $\wedge$ $a^{n-1} \equiv_n 1$

Optimierungsalgorithmus (Target-Shooting)

$G = (V, E)$

1. Lösche Knoten mit p
2. Lösche für jede verbl. Kante einen der beiden Knoten.

$\mathbb{E}[X] \geq n - np - (1-p)^2 m$

Falls $p = \frac{n}{2m} \Rightarrow \mathbb{E}[X] \geq \frac{n^2}{4m}$

Andere Idee: $Bin(n, 1-p)$; $Bin(m, (1-p)^2)$

$\#$ verbl. Knoten $\sim Bin(n, 1-p)$ —

$\#$ verbl. Kanten $\sim Bin(m, (1-p)^2) = Z$

$Z \sim |\text{Stabile Menge}|$

Chernoff:

$Pr[Z \geq (1-\delta)\mathbb{E}[Z]] \geq f(\varepsilon, n, m)$ ← p

$\max_{0\le i\le m-1} |h^{-1}(i)| \doteq \min$ für ein Datensatz

$m=n$

$\max_{0\le i\le m-1} X_i$: # Bälle im Korb i : $= (1+o(1))\left(\frac{\log(n)}{\log(\log(n))}\right)$

Power of Two Choices: $\forall$ Bälle wähle 2 Körbe zufällig und platziere Ball in geringeren Korb:

$$\max_{0\le i\le n-1} X_i = (1+o(1))\,\frac{\log(\log(n))}{\log(2)}$$

$h: \mathcal{U} \to \mathbb{Z}_n = \{0,1,\dots,m-1\}$

Datensatz $S \subseteq \mathcal{U} \wedge |S| = n$

Bloom-Filter (geht auch mit $k=1$)

k-Hashfunktionen, Speicher init 0.

Ziel: Check ob $s \in S$.

$\forall s \in S : \forall i \in \{1,\dots,k\} : M[h_i(s)] = 1$

CheckForMembership(u):

"Ja," falls $u \in S$, "Nein", sonst:

if $(M[h_1(u)] = \dots = M[h_k(u)] = 1)$

  return "Ja"; → false-positives p

else return "Nein"; → Immer richtig

Speichern $\hat{=}$ $n \cdot k$ Bälle werfen

Einsen $\hat{=}$ # nicht-leere Körbe

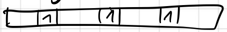
Umso grösser k, desto kleiner Fehlerwahrsch.

$Pr[\text{false positives}] \approx \left(1 - e^{-\frac{kn}{m}}\right)^k$

$k = 2,3$ Praxis

Optimales Min: $K = \frac{m}{n}\ln(2)$

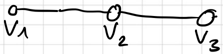
Übung Bloom-Filter



$Pr[\text{„false positive"}] \approx Pr[\text{Eintrag an } f_1(D) \text{ is } 1] \cdots Pr[\text{Eintrag an } f_k(D) \text{ ist } 1] = Pr[\text{Eintrag } f_1(D) = 1)^k = \left(1 - Pr[\text{Eintrag } f_1(D) = 0]\right)^k = \left(1 - \left(\frac{m-1}{m}\right)^{kn}\right)^K \approx \left(1 - e^{-\frac{kn}{m}}\right)^K$

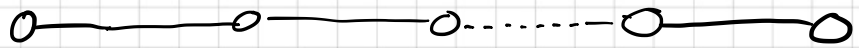
Optimal Min.: $K = \frac{m}{n}\ln(2)$

Elegant solution for problem:



$\mathbb{E}[X_2] = \frac{1}{2} \cdot 1 + \frac{1}{2}(2 + \mathbb{E}[X_2])$

$\frac{1}{2}\mathbb{E}[X_2] = \frac{1}{2} + 1 \Rightarrow \mathbb{E}[X_2] = 3$



$\mathbb{E}[X_1] = 1 + \mathbb{E}[X_2]$

$\mathbb{E}[X_i] = \frac{1}{2}(1 + \mathbb{E}[X_{i-1}]) + \frac{1}{2}(1 + \mathbb{E}[X_{i+1}])$

$\mathbb{E}[X_n] = 0$

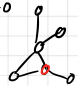
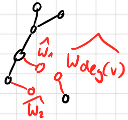
$\Rightarrow$ Generalisierbar mit Gauss-Algorithmus

Longest path in undirected graph

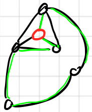
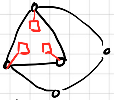
Finde Pfad mit Länge k im Graphen.

Hamiltonkreis

Konstruiere G' mit $m' \leq 2n-2$

G, wähle v:  ⇒ G': 

$n' = (n-1) + deg(v)$

 ⇒ 

Finde Pfad von Grösse n: $(u_0, u_1, \dots, u_n)$

$u_1, \dots, u_{n-1}$ sind die $n-1$ verbleibenden Knoten sein. Also muss u_0, u_n □ Knoten sein ⇒ H_K

Falls LONGPATH in $t(n)$ lösbar ⇒ H_K in $O(n^2) + t(n+n-2)$ gelöst.

Kurze lange Pfade: (G = Graph, B = Länge von Pfad):

$B = k$ lösbar in n^k (genauer $\binom{n}{k}$) $\quad B = \log(n)$

Färbung $\varphi : V \to [k] = \{1, \dots, k\}$

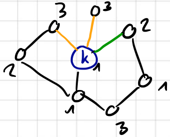
Bunter Pfad: Pfad auf gefärbten Graphen, wobei keine Doppelfarben.

Gesucht: Pfad mit Länge n.

$P_i(v) := \left\{ S \in \binom{[k]}{i+1} \;\middle|\; \text{∃ ein verbundener bunter Pfad der auf } v \text{ endet mit } S \text{ Farben. } |S| = i+1 \right\}$

=: Menge von Mengen von Farben mit Kardinalität $i+1$, aus denen ein bunter Pfad mit Länge i existiert, der auf v endet.

$P_1(k) = \{\{1,2\},\{1,3\}\}$



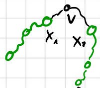
∃ bunter Pfad Länge $k-1$ $:\Leftrightarrow \bigcup_{v \in V} P_{k-1}(v) \neq \emptyset$

$$P_0(v) = \{\{\gamma(v)\}\}$$

$$P_1(v) = \{\{\gamma(x), \gamma(v)\} : x \in N(v), \gamma(x) \neq \gamma(v)\}$$

$$P_i(v) = \bigcup_{x \in N(v)} \{R \cup \gamma(v) : \gamma(v) \notin R, R \in P_{i-1}(x)\}$$

$O(i)$ Prüfen



Laufzeit P_i berechnen: $\sum_{v \in V} deg(v) \cdot \binom{k}{i} \cdot \underset{\leq k}{i} \in O\left(\binom{k}{i} \cdot i \cdot |E|\right)$

Alle Farbkomb. von Grösse i

Gesucht: Bunter Pfad mit k Farben.

$$O\Big(\underbrace{\sum_{i=1}^{k-1} \binom{k}{i}}_{2^k} \underset{\leq k}{i}\, m\Big) = O(km \cdot \Sigma) = O(2^k k \cdot m)$$

Polynomiell für $\log(v)$

Notation: $[n] = \{1, 2, \dots, n\}$; $[n]^k$ = Alle Wörter von Grösse k, $|[n]^k| = n^k$

$\binom{[n]}{k}$ = Alle k-elementige Teilmengen.

(G,B) finde Pfad von Länge B.

Färben mit $k := B+1$,

Zufällig einen Knoten wählen und färben.

Falls man einen bunten Pfad findet ⇒ Pfad B gefunden!

$P_{Erfolg} = Pr[\exists \text{ ein bunter Pfad Länge } k-1]$

$\geq Pr[\text{zufälliger Pfad } P \text{ ist bunt}]$

$$= \frac{k! \text{ Anzahl Pfade mit } k \text{ untersch. Farben}}{k^k \text{ Anzahl Pfade von Grösse } k}$$

$$\geq e^{-k} = \frac{1}{e^k}$$

Erwartete # Wiederholungen, bis ein Pfad von Länge $k-1$ (k Knoten) gefunden $= \frac{1}{P_{Erfolg}} \leq e^k$

→ Ziel $O((2e)^k k \cdot m)$ → Polynomiell

$\boxed{N = \lambda e^k}$ Wiederholungen ^ wir haben Pfad der Länge $k-1$

$Pr[\text{Erfolg in } \lambda e^k \text{ Runden}]$

$= 1 - Pr[\text{kein Erfolg in } \lambda e^k \text{ Runden}]$

$= 1 - (1-p)^{\lambda e^k} \geq 1 - (1 - e^{-k})^{\lambda e^k}$

$\geq 1 - e^{-\cancel{e^{-k}} \lambda \cancel{e^k}} = 1 - e^{-\lambda} = 1 - \frac{1}{e^\lambda}$

$O(\lambda e^k \cdot 2^k km) = O(\lambda (2e)^k km)$

Flüsse in Netzwerken

Netzwerk 5-Tupel $N=(V,A,c,s,t)$

Gerichteter Graph $G=(V,A)$ $\wedge$

Kapazitätfunktion $c: A\to\mathbb{R}_0^+ \wedge s\in V$ (Quelle) $\longrightarrow$ $t\in V$ (Sinke)

= # Datenpakete / Zeiteinheit

Ein Fluss in N ist eine Funktion $f: A\to\mathbb{R}_0^+$ mit

mit $\forall e\in A: 0\le f(e)\le c(e)$ (= Zulässigkeitseigenschaft)

$\wedge\ \forall v\in V,\ v\neq s,\ v\neq t\ \sum_{u\in V,(u,v)\in A} f(u,v) = \sum_{v\in V(u,v)\in A} f(u,v)$

Flusserhaltung: Gleich viel Daten hinein wie heraus.

Wert eines Flusses   outflow(s) − inflow(s)

$val(f) := netoutflow(s) := \sum_{u\in V:(s,u)\in A} f(s,u) - \sum_{u\in V:(u,s)\in A} f(u,s)$

inflow(s) − outflow(s)

$netinflow(q) := \sum_{u\in V:(u,q)} f(u,q) - \sum_{u\in V:(u,q)} f(q,u)$

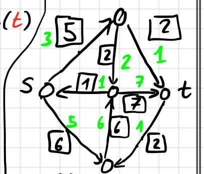
Lemma 1 $val(f) := netinflow(t)$

Per Definition Fluss:

$0 \overset{!}{=} \sum_{(v,u)\in A} f(v,u) - \sum_{(u,v)\in A} f(u,v) =$

$= \sum_{v\in V}\underbrace{\left(\sum_{u:(v,u)\in A} f(v,u) - \sum_{u:(u,v)\in A} f(u,v)\right)}_{=0 \text{ für } v\notin\{s,t\}}$

$= \underbrace{\sum_u f(s,u) - \sum_u f(u,s)}_{=\text{netoutflow}(s)} + \underbrace{\sum_u f(t,u) - \sum_u f(t,u)}_{=\text{netoutflow}}$



$val(f) = 3+5-1 = 7$

$netinflow(t) = 7$

Bei Diagrammen ist Kapazität in Box $\boxed{c}$

Fluss negativ ⇒ Senke nach Quelle

Schnitte

Welche Rohre zerstören um Fluss zu minimieren.

s-t Schnitt $:\equiv$ $V = S \uplus T$, $s\in S$, $t\in T$

Wenn keine Kante im Schnitt von $S\to T$ kein Fluss.

Kapazität des Schnittes $:= \sum_{(u,w)\in(S\times T)\wedge A} c(u,w) =: cap(S,T)$

ACHTUNG: Ignoriert Kanten von $T\to S$.

Lemma 2: $val(f) \le cap(S,T)$

Lemma 3:

$val(f) = f(S,T) - f(T,S) \le f(S,T) \le cap(S,T)$

$f(S,T) := \sum_{u\in S, v\in T} f(u,v)$ $f(X,Z)+f(Y,Z) = f(X\cup Y, Z)$

Beweis: Wenn man $f(S,T) = cap(S,T)$ findet ⇒

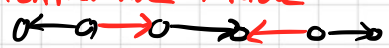
Satz Max-Flow / Min-Cut Theorem

$N=(V,A,c,s,t)$ ⇒ Gut zum Beweisen!

$\max_f val(f) = \min_{(S,T)} cap(S,T)$

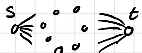
∞ Flüsse $\le 2^N$ Schnitte

Augmentierende Pfade   Zurück zu den Flüssen

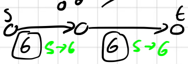


Intuition

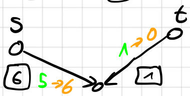
1. Finde Pfad von s-t Richtungegal.



2. Erhöhe alle Kantenflüsse um Kap. (oder neg falls ←)



Gibt es keine Verbesserung mehr ⇒ Max-Flow!



Restnetzwerk

Geg. $N \wedge f \Rightarrow N_f$

Wobei: N ohne Kanten in beide Richtungen:

$N_f = (V, A_f, r_f, s, t)$

e^{opp} = entgegengesetzte Kante in N

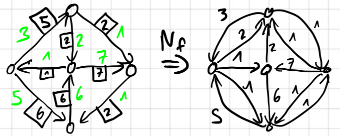
Alle Kanten in N_f: ($r_f :=$ Restkapazität)

① $e\in A \wedge f(e) < c(e) \Rightarrow e\in A_f \wedge r_f := c(e) - f(e)$

② $e\in A \wedge f(e) > 0 \Rightarrow e_{opp}\in A_f \wedge r_f(e^{opp}) = f(e)$

$\sum_e f(e) = 0 \vee \sum_e f(e) = \sum_e cap(e) \Rightarrow$ Keine Rückwärtskanten. $(r_f(e) + r_f(e^{opp})) = cap(e)$

Ein Fluss f ist maximal ⇔ ∄ Pfad von s nach t im N_f



Finde Pfad von s-t.

Jeder maximale Fluss hat einen entspr. min. Schnitt.

Für rationale Kapazitäten (multiplizieren mit KGV) endlich lang.

Algorithmus Beweis: Finde Pfad von s-t in N_f und $\varepsilon = \min(P)$

$$f'(e) = \begin{cases} f(e)+\varepsilon & \text{falls } e\in P \\ f(e)-\varepsilon & \text{falls } e^{opp}\in P \\ f(e) & \text{sonst} \end{cases}$$

Maximalitätsbeweis: Angenommen N_f hat keinen s-t Pfad.

$S :=$ Menge aller von s aus erreichbare Knoten

$T = V \setminus S$

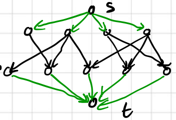
$e=(u,v)\in(S\times T)\wedge A \Rightarrow e\notin A_f \Rightarrow f(e)=c(e)$

$e=(u,v)\in(T\times S)\wedge A \Rightarrow e^{opp}\notin A_f \Rightarrow f(e)=0$

$val(f) = f(S,T) - f(T,S) = cap(S,T) - 0$ □

Ford-Fulkerson $cap \in \mathbb{N}$: Laufzeit: $O(N \cdot \underset{\uparrow}{U} \cdot M)$
$\exists$ immer ganzzahliger Max-Flow! grösste Kapazität

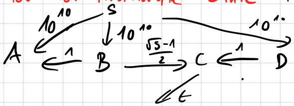
Für bipartiter Graph



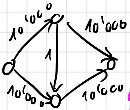
1. Richte Kanten in eine der Mengen
2. Füge s und t hinzu.
3. Kapazitäten = 1
4. Da in jeder Kante höchstens Fluss 1 hat (sonst 0) $\exists$ matching.

Augmentierende Pfade (Im N_f gilt gematchte Kanten in die andere Richtung gehen).

Flow für irrationale Zahle FORD-FULKERSON WIKIPEDIA



Multiplying all capacities with λ, also multiplies all capacities. $\sum \lambda\, cap(e) = \lambda \sum cap(e)$



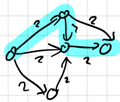
"Fulkerson Killer"

A maximum capacity path can be found by either using Dijkstra (but reverse property) or by finding the maximal spanning tree and find any path between s and t.

Removing e if in S,T min cut will surely decrease cut by $c(e) \Rightarrow |f'| = |f| - c(e)$

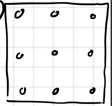
Adding 1 to capacity (can change flow only if $f(u,v) = c(u,v)$) $\rightarrow$ New path in N_f

Decreasing 1 from capacity (u,v). Find path in N_f from $s \rightarrow u$ and $t \rightarrow v$. Then increase flow in direction $t \rightarrow (v,u) \rightarrow s$. Now retry.

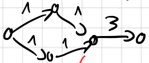


Find a star
$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

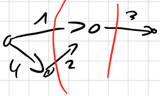
Bildsektion (Vordergrund / Hintergrund)



β_p Hintergrund α_p Vordergrund
$\alpha_p > \beta_p$: Vordergrund | $\beta_p > \alpha_p$: Hintergrund
$\hookrightarrow$ Zu stark granuliert. Zwischen u,v Strafwert γ



hat |Max Flow| = 2



hat 2 Mincuts

P := Menge von Pixel, $\underset{\text{Kanten}}{E} \subseteq \binom{P}{2}$ (Nachbarschaftbez.)

α_p := Erwartung P im Vordergrund
β_p := Erwartung P im Hintergrund
Vordergrund $A := \{p \in P \mid \alpha_p > \beta_p\}$
Hintergrund $B := P \setminus A$

Problem: zu feinkörnig

Lösung: Kanten kosten von $e \in E$ = Strafzoll $\gamma_e = 1$

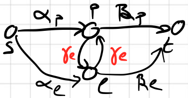
Qualitätsmass $q(A,B) = \sum_{p \in A} \alpha_p + \sum_{p \in B} \beta_p - \sum_{\substack{e \in E \\ |e \cap A| = 1}} \gamma_e$

$q(A,B) := \overbrace{\sum_{p \in P} \alpha_p + \beta_p}^{Q} - \sum_{p \in A} \beta_p - \sum_{p \in B} \alpha_p - \sum_{e \in E} \gamma_e$

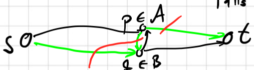
$q'(A,B) := \sum_{p \in A} \beta_p + \sum_{p \in B} \alpha_p + \sum_{\substack{e \in E \\ \text{Kanten } A-B}} \gamma_e$

$\max q = \min q'$ $\Rightarrow$ min cut

$A \uplus B$ Partitionierung P



Minimaler Schnitt:
Falls $p \in B$ $(s,p) \in$ Schnitt
Falls $p \in A$ $(p,t) \in$ Schnitt
Falls $p \in A \wedge q \in B$ $(p,q) \in$ Schnitt



$= \min q' = \max q$

$cap(S,T) = \sum_{p \in B} \alpha_p + \sum_{p \in A} \beta_p + \sum_{(p,q) \in E \wedge p \in A,\, q \in B} \gamma_{p,q}$

Konvexe Menge $N = (V, E, c, s, t)$
$\mathbb{O}$ = Nullfluss : $A \rightarrow \mathbb{R}$ mit $\mathbb{O}(e) = 0$ ist immer Fluss!
Sei nun $f_1 \neq \mathbb{O}$ ein weiterer Fluss:
$f_1^* : A \rightarrow \mathbb{R}$ $\forall e \in E \left(f_1^*(e) := \frac{1}{2} f_1(e) \right)$
$\hookrightarrow$ Alle eingehende Flüsse & ausgehende sind gleich.

Lemma: f_v, f_w Flüsse $\in N \Rightarrow f_\lambda(e) = (1-\lambda) f_v(e) + \lambda f_w(e)$
$\wedge\ 0 < \lambda < 1$

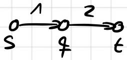
Beweis Zulässigkeit: $(1-\lambda)\underset{\leq c(e)}{f_v(e)} + \lambda \underset{\leq c(e)}{f_w(e)} \leq (1-\lambda) c(e) + \lambda c(e) = c(e)$

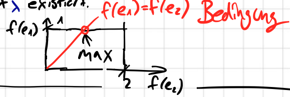
Korrolar: $val(f_\lambda) = (1-\lambda)\, val(f_v) + \lambda\, val(f_w)$
Korrolar: Entweder nur $\mathbb{O}$ oder ∞-Flüsse.
$(e_1, \dots, e_{|A|=m})$ = Ordnung der Kanten von A.
$f_{\dots} : A \rightarrow \mathbb{R}$ $(f(e_1), \dots, f(e_m)) \in \mathbb{R}^m$

Liniensegment:
$\overline{V_0 V_1} := \{ (1-\lambda) V_0 + \lambda V_1 \mid \lambda \in [0,1] \}$

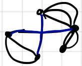
Eine Menge heisst konvex, falls $\overline{V_0 V_1} \in C \subseteq \mathbb{R}^m$

Die Menge der Flüsse bildet konvexe Menge $\in \mathbb{R}^m$
" " maximalen Flüsse " " " $\in \mathbb{R}^m$
Also jeder Fluss f_λ existiert.





Minimale Schnitte $G = (V, E)$ ungewichtet, schleifenlos MULTIGRAPH!
$C \subseteq E$ = Kantenschnitt
$G' = (V, E \setminus C)$ nicht zusammenhängend

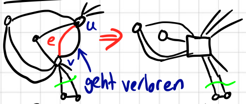


$\mathcal{N}(G) = 2$ = Min. # Kantenschnitte $\overset{\text{entferne alle}}{\leq} \delta(G)$

Probleme:
- Gewicht simpler Graph = Anzahl
- Wir wissen nur S-T Schnitt
  $\rightarrow$ Wähle ein s und für alle $t \in V \setminus \{s\}$
- Ungerichtet $\Rightarrow$ 2 gerichtete Kanten $\Rightarrow$ 3 Kanten + extra v'

$O(n \cdot \underbrace{n\, m \log(n) + m}_{\text{Flusslaufzeit}})$

Kantenkontraktion (2 Knoten zusammen)



Schnitte die $u \in S, v \in T$ Kanten haben, gehen verloren.

$G \rightsquigarrow G/e$ (Operation) geht in $O(n)$

Algorithmus cut(G) Karger algorithmus $O(n^2)$

```
while (|V| > 2) {
    e = uniform(E); // in O(n)
    G = G/e; // in O(n)
} return μ(G); // Anzahl Kanten zw. u, v
```

$deg_G(v) = \#$ inzidente Kanten # Kanten u-v

$deg_{G/e}(u,v) = deg_G(u) + deg_G(v) - 2k$

$\mu(G/e) \geq \mu(G)$

Falls Schnitt $|C| = \mu(G) \wedge e \notin C \Rightarrow \mu(G/e) = \mu(G)$

$G = (V,E)$ Multigraph, $|V| = n$, e uniform E.

$\Rightarrow Pr[\mu(G) = \mu(G/e)] \geq 1 - \frac{2}{|V|}$

C ein minimaler Schnitt $k := \mu(G) = |C|$

$\Rightarrow$ Jeder Knoten hat Grad mind. k $\Rightarrow$ Sonst alles um v schneiden.

$\Rightarrow |E| = \frac{1}{2} \sum_{v \in V} \underbrace{deg_G(v)}_{\geq k} \geq \frac{1}{2}|V| \cdot k$

Start here $\Rightarrow Pr[\mu(G) = \underbrace{\mu(G/e)}_{\text{eine Kontrakt.}}] \geq Pr[\underbrace{e \notin C}_{\text{e nicht im Min. Schnitt}}] = 1 - \frac{|C|}{|E|}$

$\geq 1 - \frac{k}{\frac{1}{2}|V|k} = 1 - \frac{2}{|V|}$ $\square$ $1 - \frac{|C|}{|E|}$

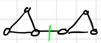
$\hat{p}(G) := Pr[cut(G) = \mu(G)]$

$\hat{p}(n) := \inf_{\substack{G=(V,E) \\ |V|=n}} \hat{p}(G)$ (unendlich viele Zusammenhäng.. Multigraphen) mit n Knoten

$\hat{p}(2) = 1$ (Gibt immer das richtige aus) $\hat{p}$(10, 10, 10) $= 1$

$\hat{p}$(200, 800, 500) $= \frac{800}{1500}$ $\hat{p}(G) =$ (301, 300, 300) $= \frac{301}{901}$

$\hat{p}(3) = \frac{1}{3} \to$ konvergiert



Lemma: $\hat{p}(n) \geq \left(1 - \frac{2}{n}\right)\hat{p}(n-1)$ | $\hat{p}(3) \geq 1/3$ per inf, $\hat{p}(4) \geq 1/2 \cdot 1/3 = 1/6$

$E_1 :=$ Ereignis $\mu(G/e) = \mu(G)$

$E_2 :=$ Ereignis $CUT(G/e) = \mu(G/e)$

$\hat{p}(G) = Pr[E_1 \wedge E_2] = \underbrace{Pr[E_1]}_{\geq 1 - \frac{2}{n}} \cdot \underbrace{Pr[E_2 | E_1]}_{\geq \hat{p}(n-1) \text{ mind.}}$ $\square$

Die obere Schranke kann durch Beispiele gegeben werden.

$\hat{p}(n) \geq \frac{n-2}{n}\hat{p}(n-1) \geq \frac{n-2}{n}\frac{n-3}{n-1}\hat{p}(n-2) \ldots$

$\geq \frac{1}{\cancel{3}} \cdot \frac{2}{\cancel{4}} \cdot \frac{\cancel{3}}{\cancel{5}} \cdots \frac{\cancel{n-4}}{\cancel{n-2}} \cdot \frac{\cancel{n-3}}{n-1} \cdot \frac{\cancel{n-2}}{n} = \frac{2}{n(n-1)} = \frac{1}{\binom{n}{2}}$

$\hat{p}(n) \geq \frac{1}{\binom{n}{2}} \Rightarrow$ Erwartungswert $\leq \binom{n}{2}$

$\lambda\binom{n}{2}$ - Wiederholungen ist korrekt mit $\geq 1 - e^{-\lambda}$

$\lambda = \ln(n) \Rightarrow 1 - e^{-\ln(n)} = 1 - \frac{1}{n}$ | $O(n^2\binom{n}{2}\log(n)) = O(n^4\log(n))$

Der Algorithmus geht gut bis 3 Knoten.

Im letzten Schritt ist Fehlerwahrsch. am grössten.

n Knoten $\xrightarrow{\text{random Kontrak.}}$ t Knoten $O(t^4\log(t))$ deterministisch

$O((n-t) \cdot n) = O(n^2)$

Erfolgswahrsch. von diesem $O(n^2 + t^4\log(t))$ Algorithmus.

$\geq \frac{\cancel{n-2}}{n} \cdot \frac{\cancel{n-3}}{n-1} \cdots \frac{t}{\cancel{t+2}} \cdot \frac{t-1}{\cancel{t+1}} \underbrace{\hat{p}(t)}_{=1 \text{ (da deterministisch)}}$ oder < 1 für probab.

$\geq \frac{t(t-1)}{n(n-1)}$ Erwartete Wiederholungen: $\frac{n(n-1)}{t(t-1)}$

Bei $\lambda\frac{n(n-1)}{t(t-1)}$ Wiederholungen gilt Erfolg $> 1 - e^{-\lambda}$

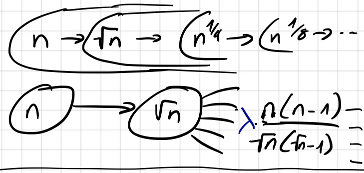
Laufzeit: $O\left(\lambda\frac{n(n-1)}{t(t-1)}(n^2 + t^4\log(t))\right)$

$O\left(\frac{n^4}{t^2} + n^2t^2\log(t)\right)$ wähle $t = \sqrt{n}$ $t^3\log(t)$

ABER wir haben nun Laufzeit $O(n^3\log(n))$

Bootstrapping

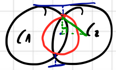
Diesen Algorithmus kann man nun rekursiv einsetzen und kann den Grenzwert finden.



Konvexe Hülle 2D / Voronoi Diagramme (Nächster Starbucks)

$C(P)$ = Kleinster umschliessender Kreis 2D (eindeutig)

Beweis ↯ Mittelpunkte verschieden aber gleichen Radius.



$P \subseteq C_1 \cap C_2 \subseteq \hat{C}$ hat aber kleineren Radius.

$CB(Q)$ = Kleinster umschl. Kreis der auf allen Punkten Q liegt. Enthält mind. 2 Punkte auf Rand.

Wenn nur 2 Punkte: $\{\frac{1}{2}(M_x - m_x) + m_x, \frac{1}{2}(M_y - m_y) + m_y\}$

Wenn 3 Punkte: Mittelsenkrechte von

$\Rightarrow \exists Q \subseteq P$ mit $2 \leq |Q| \leq 3 : C(P) = CB(Q)$

$O(N^4)$ Brute-Force Genau: $\left(\binom{n}{2} + \binom{n}{3}\right) \cdot n$

$\to$ Finde Punkt der zw. P_i und P_j liegt.

Randomisierter Algorithmus $Q = \text{uniform} \binom{[P]}{3} \cup \binom{[P]}{2}$

Erwartete Laufzeit: $\frac{1}{\frac{1}{\binom{n}{2} + \binom{n}{3}}} \cdot n \in \Theta(n^4)$

3 Punkte aufeinander = ignorieren. $\frac{\binom{n-3}{9}}{\binom{n}{12}}$

Nehme $C(Q)$, wobei $|Q| = 12$

```
if (C(Q) alle Punkte enth.) return Q
else    verdopple alle Punkte nicht in Q
```

$n_i = \#$ Kopien von Punkt P_i

Bestimme Q zufällig $r \in \{1,\dots,N\}$, $N = \sum n_i$

$|Q| = 12$

Kleinste j, sodass $\sum_{i=1}^{j} n_i \geq r$ — $O(n)$ reicht! ↳ Da man später sowieso alle anschaut

$(12 \cdot N + N) \cdot T$

↓ j bestimmen ↓ Kollisionen ↓ # Iterationen

$X_k = \sum$ aller Punkte nach k Iter.

$T \geq k$: $X_k \geq 2^{k/3}$ denn mind. ein Punkt aus $Q \geq \frac{k}{3}$ mal ausserhalb.

↑ Grösse

$\mathbb{E}[\#\text{Punkte ausserhalb von } C(Q)] \leq \frac{3}{|Q|+1}\cdot n$

$\mathbb{E}[X_k] \leq (1+\frac{3}{13})^k \cdot n \quad \leq (1+\frac{3}{13})t$

Beweis: $\mathbb{E}[X_k] = \sum_t [X_k \mid X_{k-1}=t]\, Pr[X_{k-1}=t]$

$= \sum_t (1+\frac{1}{13})\, t \; Pr[X_{k-1}=t]$

$=(1+\frac{1}{13})\, \mathbb{E}[X_{k-1}=t]$

$\leq (1+\frac{1}{13})^2\, \mathbb{E}[X_{k-2}=t] \leq (1+\frac{1}{13})^k\, \underbrace{\mathbb{E}[X_0=t]}_{=n}$

$(1+1/13)^k \nearrow$ als $2^{k/3}$

$2^{k/3} \cdot \underbrace{Pr[T \geq k]}_{\leq 1} \leq \mathbb{E}[X_k] \leq (1+\frac{3}{13})^k\, n$

$\Rightarrow Pr[T \geq k] \leq 0.98^k \cdot n$

$\mathbb{E}[T] = \sum_k Pr[T \geq k] = -\log_{0.98}(n) + O(1) \leq 50 \cdot \ln(n)$

P sind n Punkte (Multimenge), $r \in \mathbb{N}$, $R \in \binom{P}{r}$ zufällig.

$X = \#$ Punkte ausserhalb von $C(R)$

$\mathbb{E}[X] \leq \frac{3(n-r)}{r+1}$ für $r = \sqrt{n} \sim \mathbb{E}[X] \in O(\sqrt{n})$

$out(\underset{\text{Punkt}}{p}, \underset{\text{Menge von Punkten}}{R}) = \begin{cases} 1, & p \notin C(R) \\ 0, & \text{sonst} \end{cases}$

$X = \sum_{p \in P\setminus R} out(p,R) \Rightarrow \mathbb{E}[X] = \sum_{p \in P\setminus R} \mathbb{E}[out(p,R)]$

$essential(\underset{\text{Punkt}}{p}, \underset{\text{Menge von Punkten}}{Q}) = \begin{cases} 1, & C(Q\setminus\{p\}) \neq C(Q) \\ 0, & \text{sonst} \end{cases}$

$out(p,R) \Leftrightarrow essential(p, R \cup \{p\}) = 1$

$\mathbb{E}[X] = \frac{1}{\binom{n}{r}} \sum_{R \in \binom{P}{r}} \sum_{s \in P\setminus R} out(s,R)$

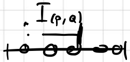
$\overset{*}{=} \frac{1}{\binom{n}{r}} \sum_{R \in \binom{P}{r}} \sum_{s \in P\setminus R} essential(s, R \cup \{s\})$

r-1 hinzufügen | r+1 rausrechnen $\overset{*}{\;} \sum_{Q \in \binom{P}{r+1}} \underbrace{\sum_{s \in Q} essential(s,Q)}_{\leq 3}$

Höchstens 3 auf Rand mit Einfluss!

$\leq \frac{1}{\binom{n}{r}} \sum_{Q \in \binom{P}{r+1}} 3 \leq \frac{\binom{n}{r+1}}{\binom{n}{r}} \cdot 3 = 3\,\frac{n-r}{r+1}$

n Punkte in $\mathbb{R}^1$ verschieden



$\mathbb{E}[\#\text{Punkte ausserhalb von } I] = 2 \cdot \frac{n-2}{2+1}$

↓ Wobei 2 Punkte

Falls $\sqrt{n}$ Punkte aus $n \Rightarrow 2 \cdot \frac{n-\sqrt{n}}{\sqrt{n}+1}$

$\mathbb{E}[\#\text{Punkte ausserhalb von achsenparallelen}] \leq 4\,\frac{n-r}{r+1}$

$\Rightarrow essential(Q,p) = 0 \;\; \forall p \in Q$

Kugel in D-Dimensionen braucht D+1 Punkte.

Geht nicht im umschliessenden Dreieck

⇒ lässt sich drehen und deswegen $\sum_{s \in Q} essential(s,Q) = n$

Ellipsoid oder konvexe Hülle geht für Roboter.

p auf Gerade durch q und r: $\lambda \in \mathbb{R}$

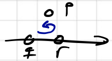
$p = \lambda_1 q + \lambda_2 r \wedge \lambda_1 + \lambda_2 = 1 \Rightarrow p = (1-\lambda) q + \lambda r$

$\det \begin{vmatrix} p_x & p_y & 1 \\ q_x & q_y & 1 \\ r_x & r_y & 1 \end{vmatrix} = 0 \Leftrightarrow p$ auf Geraden p,q ⇔ Eine Reihe ist kollinear.

Gerichtete Gerade von (q,r). Wenn >0 links. <0 rechts.

Falls $q = r \Leftrightarrow$ Det immer 0.

Gegenuhrzeigersinn p,q,r

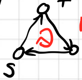


>0 + Sinne

$\mathbb{R}^3$ $\begin{pmatrix} p_x & p_y & p_z & 1 \\ q_x & q_y & q_z & 1 \\ r_x & r_y & r_z & 1 \\ s_x & s_y & r_y & 1 \end{pmatrix}$ p auf Ebene (q,r,s)

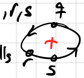
$>0 \Leftrightarrow$ Positive Seite

$<0 \Leftrightarrow$ Negative Seite



Negativ von uns aus.

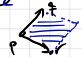
$\det \begin{vmatrix} p_x & p_y & p_x^2+p_y^2 & 1 \\ q_x & q_y & q_x^2+q_y^2 & 1 \\ r_x & r_y & r_x^2+r_y^2 & 1 \\ s_x & s_y & s_x^2+s_y^2 & 1 \end{vmatrix} = 0 \Leftrightarrow p$ auf Kreis q,r,s; $>0 \Leftrightarrow p$ im Kreis, falls



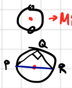
Punkt testen in Mitte in r∧s Trick.

Oder Determinante

Fläche $|\det(p,q,r)|$ =



Kreis durch 2 Punkte



$\langle q-p, r-q \rangle = 0$

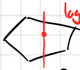
Konvexe Hülle 2D

$conv(S) := \bigcap_{\substack{S \subseteq C \subseteq \mathbb{R}^d \\ C \text{ konvex}}} C$ (Alle Punkte in konvexer Hülle)

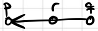
Wenn 2 Punkte in konvexer Menge ⇒ Segment in konv. Hülle.

Randkante $\vec{pq}$:⇔ Wenn alle Punkte auf einer Seite liegen (oder auf).

log(n) in Hülle



Kanten (ungerichtet) & Knoten/Ecken

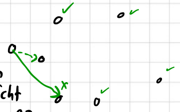


$\vec{pq}$ ist Randkante, $\vec{qr}$ nicht!

Konvexe Hülle (Punkt am weitesten links, falls gleich oberste)

Jarvis - Wrap

Nimm eine beliebige Kante und schau für alle Punkt ob sie links sind. Falls nicht, wähl einen anderen Punkt (Punkte müssen nicht neu überprüft werden). Geht bis ganz am Ende.



$O(n \cdot h)$ (wobei h = Anzahl Kanten im konv. Polygon)

$\in O(n^2)$

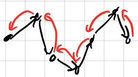
Sortieren um Duplikate zu detektieren $O(n^2 \log(n))$

Robustheit ungenau für $\sqrt{2}$

$(q_x - p_x)\cdot(r_y - p_y) \gtreqless (q_y - p_y)\cdot(r_x - p_x)$

Floating-point filter

Lokale Reperatur



1) Sortiere Punkte
2) Gehe über das Array und so lange 3 nacheinander folgende Punkte die falsche Orientation haben, lösche mittleren. (Stack implementiert werden).

$2n-2-h$ erfolgreiche Tests $O(n)$ ohne sortieren.

$+\ 2(n-1)$ unerfolgreiche Tests (maximal ausser links/rechts)

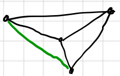
$4n-4-h$ (nette amortisierte Analyse)

$h \approx$ # edges on convex hull $O(\sqrt[3]{n})$

$\approx$ # edges on rectangle $O(\log(n))$

Gut für uniform in Kreis:

$O(\log(n))$ Test für Punkt in konvexe Hülle.



In konvexe Hülle $\Rightarrow$ Nichts

Ausserhalb " " $\Rightarrow$ Repariere

$O(n\log(n))$ wegen Schneiden.

$O(n \sum_{i=1}^{n} \frac{f(r+1)}{r(r+1)})$ f = Erwartete # Ecken auf $R \in \binom{P}{r}$

Auf Kreisumfang worst-case: $f(x) = x$

Auf Kreis worst-case: $f(x) = 3\sqrt[3]{x}$

$$n \sum \frac{1}{r^{5/3}} \in O(n)$$

Lazy evaluation on polynomial continuous fractions!

Wähle t Zahlen aus D mit je Wahrsch. d_i

$O(n + t\log(n))$ Präfixsumme von D.

$O(n+t)$ Radix sort über t und dann mit t einmal drüber laufen.

Leda for numeric integration

Zusatz:

- Anti-Kanten $\Rightarrow \exists$ Fluss mit $e=0$ oder $e'=0$
- Maximale Anzahl Kantendisjunkte Pfade von $s \to t$ erhält man mit 0/1 flow.

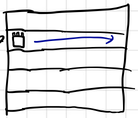
- Maximale Knotendisjunkte Pfade erhält man wenn man jeden Knoten  aufteilt zu 

 • Teile Netz in Source ∧ Sink ein.

Türme auf Schachbrett.

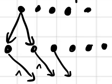
Platziere Turm →

1. Wähle Reihe



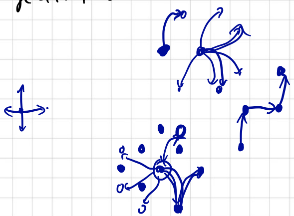
Reihe:

Spalte:



Höchstens 1 Turm pro Reihe.
Höchstens 1 Turm pro Spalte.
⇒ Bipartites Matching, Kante von Reihe zu Spalte = Turm.

- Minimum Kapazität kann mit schon markierten Fluss gehalten werden.



→ Nach finden reduzieren einer Kapazität