Algorithmen und Wahrscheinlichkeit

Nicola Studer
nicstuder@student.ethz.ch

20. Juni 2022

1 Graphen

1.1 Terminologie

- K_n := Vollständiger Graph mit n Knoten
- C_n := Kreisgraph mit n Knoten
- P_n := Pfad mit n Knoten
- H_d := d-dimensionaler Hyperwürfel
- Hamiltonkreis := Ein Kreis in G, der jeden Knoten genau einmal enthält. $\mathcal{O}(n^2 2^n)$
- Eulertour := Ein geschlossener Weg in G, der jede Kante genau einmal enthält

1.2 Zusammenhang

Def 1.23 (k-zusammenhängend)**.** Ein Graph $G = (V, E)$ heisst *k-zusammenhängend*, falls $|V| \geq k + 1$ und für alle Teilmengen $X \subseteq V$ mit $|V| < k$ gilt: Der Graph $G[V \setminus X]$ is zusammenhängend.

Def 1.24 (k-kanten-zusammenhängend)**.** Ein Graph $G = (V, E)$ heisst *k-kanten-zusammenhängend*, falls für alle Teilmengen $X \subseteq E$ mit $|X| < k$ gilt: Der Graph $(V, E \setminus X)$ is zusammenhängend.

Satz 1.25 (Menger)**.** Sei $G = (V, E)$ ein Graph. Dann gilt:

a) G ist k-zusammenhängend $\iff \forall u, v \in V, u \neq v$ gibt es k intern-knotendisjunkte u-v-Pfade-Pfade

b) G ist k-kanten-zusammenhängend $\iff \forall u, v \in V, u \neq v$ gibt es k kantendisjunkte u-v-Pfade

Bmk. (Knoten-) Zusammenhang $\leq$ Kanten-Zusammenhang $\leq$ minimaler Grad

Bmk (low-Werte)**.**

$$low[v] = \min\left(dfs[v], \min_{(v,w)\in E}\begin{cases} dfs[v] & \text{if } (v,w) \text{ rest-edge} \\ low[w] & \text{if } (v,w) \text{ tree-edge}\end{cases}\right)$$

Artikulationsknoten. Sei $G = (V, E)$ ein zusammenhängender Graph. $v \in V$ Artikulationsknoten $\iff G[V \setminus \{v\}]$ nicht zusammenhängend. Artikulationsknoten, wenn:

1. $v \neq$ root und v hat Kind u im DFS-Baum mit low$[u] \geq$ dfs$[v]$
2. $v =$ root und v hat mindestens zwei Kinder im DFS-Baum.

Brücken. $e \in E$ Brücke $\iff G - e$ nicht zusammenhängend. Eine Baumkante $e = (v, w) \in E$ ist genau dann eine Brücke, wenn low$[w] >$ dfs$[v]$. Restkanten sind niemals Brücken.

Lemma. *Sei $G = (V, E)$ ein zusammenhängender Graph. Ist $\{x, y\} \in E$ eine Brücke so gilt:* $\deg(x) = 1$ *oder x ist Artikulationsknoten.*

Satz 1.28. Für zusammenhängende Graphen $G = (V, E)$, die mit Adjazenzlisten gespeichert sind, kann man in Zeit $\mathcal{O}(|E|)$ alle Artikulationsknoten und Brücken berechnen.

Def 1.29. Sei $G = (V, E)$ ein zusammenhängender Graph. Für $e, f \in E$ definieren wir eine Äquivalenzrelation durch:

$$e \sim f :\iff e \begin{cases} e = f, & \text{oder} \\ \exists \text{Kreis durch } e \text{ und } f \end{cases}$$

1.3 Kreise

Satz 1.31. Ein zusammenhängender Graph $G = (V, E)$ enthält eine Eulertour $\iff$ der Grad jedes Knotens gerade ist. Die Tour kann man in $\mathcal{O}(|E|)$ Zeit finden.

Satz 1.32. Seien $m, n \geq 2$. Ein $n \times m$ Gitter enthält einen Hamiltonkreis $\iff n \cdot m$ gerade ist.

Satz 1.40 (Dirac 1952)**.** Jeder Graph $G = (V, E)$ mit $|V| \geq 3$ und Minimalgrad $\delta(G) \geq \frac{|V|}{2}$ enthält einen Hamiltonkreis.

Für das Metrische Traveling Salesman Problem gibt es einen 2-Approximationsalgorithmus mit Laufzeit $\mathcal{O}(n^2)$.

1.4 Matchings

Matching. Eine Kantenmenge $M \subseteq E$ heisst Matching in einem Graphen $G = (V, E)$, falls kein Knoten des Graphen zu mehr als einer Kante aus M inzident ist.

$$e \cap f = \varnothing \text{ für alle } e, f \in M \text{ mit } e \neq f$$

Ein Knoten wird von M überdeckt, falls es eine Kante $e \in M$ gibt, die v enthält.

Perfekts Matching. Ein Matching M heisst perfektes Matching, wenn jeder Knoten durch genau eine Kante aus M überdeckt wird, oder, anders ausgedrückt, wenn $M = \frac{|V|}{2}$

Matching Typen.

- M heisst inklusionsmaximal, falls gilt $M \cup \{e\}$ ist kein Matching für alle Kanten $e \in E \setminus M$.
- M heisst kardinalitätsmaximal, falls gilt $|M| \geq |M'|$ für alle Matchings M' in G.

Satz 1.47. Der Algortihmus Greedy-Matching bestimmt in Zeit $\mathcal{O}(|E|)$ ein inklusionsmaximales Matching M_{Greedy} für das gilt:

$$|M_{Greedy}| \geq \frac{1}{2}|M_{max}|$$

wobei M_{max} ein kardinalitätsmaximales Matching sei.

Augmentierender Pfad. Ein M-augmentierenderPfad ist ein Pfad, der abwechselnd Kanten aus M und nicht aus M enthält und der in von M nicht überdeckten Knoten beginnt und endet.

$\implies$ durch tauschen entlang M können wir das Matching verbessern.

Satz 1.48 (Berge)**.** Ist M ein Matching in einem Graphen $G = (V, E)$, das nicht kardinalitätsmaximal ist, so existiert ein augmentierender Pfad zu M.

Satz 1.51. Für das Metrische Travelling Salesman Problem gibt es einen 3/4-Approximationsalgorithmus mit Laufzeit $\mathcal{O}(n^3)$ mit MST, Matching und Eulertour.

Satz 1.52 (Hall, Heiratssatz)**.** Ein bipartiter Graph $G = (A \uplus B, E)$ enthält ein Matching M der Kardinalität $|M| = |A| \iff \forall X \subseteq A\ (|X| \leq |N(X)|)$

Cor (Frobenius)**.** *Für alle k gilt: Jeder k-reguläre bipartite Graph enthält ein perfektes Matching.*

1.5 Färbungen

Def 1.56. Eine Färbung eines Graphen $G = (V, E)$ mit k Farben ist eine Abbildung $c : V \to [k]$, so dass gilt

$$c(u) \neq c(v) \quad \text{für alle Kanten } \{u, v\} \in E$$

Die chromatische Zahl $\chi(G)$ ist die minimale Anzahl Farben, die für eine Knotenfärbung von G benötigt wird.

$$\chi(G) \leq k \iff G \text{ } k\text{-partit}$$

Satz 1.58. Ein Graph $G = (V, E)$ ist genau dann bipartit, wenn er keinen Kreis ungerader Länge als Teilgraphen enthält.

Satz 1.59 (Vierfarbensatz)**.** Jede Landkarte lässt sich mit vier Farben färben.

Bmk.

- Die Heuristik findet immer eine Färbung mit 2 Farben für Bäume
- ist ein Graph planar (Kann überkreuzungsfrei in der Ebene gezeichnet werden), so gibt es immer einen Knoten vom Grad ≤ 5.
- Die Heuristik findet eine Färbung mit ≤ 6 Farben für planare Graphen
- $G = (V, E)$ zshgd. und es gibt $v \in V$ mit $\deg(v) < \Delta(G)$. Heuristik (Breiten/Tiefensuche) liefert Reihenfolge, für die der Greedy-Algorithmus höchstens $\Delta(G)$ Farben benötigt.

Satz 1.60. Sei G ein zusammenhängender Graph. Für die Anzahl Farben $C(G)$, die der Algorithmus Greedy-Färbung benötigt, um die Knoten des Graphen G zu färben, gilt

$$\chi(G) \leq C(G) \leq \Delta(G) + 1$$

ist der Graph als Adjazenzliste gespeichert, findet der Algortihmus die Färbung in zeit $\mathcal{O}(|E|)$

Cor. *Ist G ein Graph, in dem man jeden Block mit k Farben färben kann, dann kann man auch G mit k Farben färben.*

Theorem. $\forall k \in \mathbb{N}, \forall r \in \mathbb{N}$: es gibt Graphen ohne einen Kreis mit Länge $\leq k$, aber mit chromatischer Zahl $\geq r$.

Satz 1.64 (Brooks)**.** Ist $G = (V, E)$ ein zusammenhängender Graph, $G \neq K_n, G \neq C_{2n+1}$, so gilt:

$$\chi(G) \leq \Delta(G)$$

und es gibt einen Algorithmus, der die Knoten des Graphen in Zeit $\mathcal{O}(|E|)$ mit $\delta(G)$ Farben färbt.

Satz 1.66 (Mycielski-Konstruktion)**.** Für alle $k \geq 2$ gibt es einen dreiecksfreien Graphen G_k mit $\chi(G_k) \geq k$.

Satz 1.67. Einen 3-färbbaren Graphen kann man in Zeit $\mathcal{O}(|V| + |E|)$ mit $\mathcal{O}(\sqrt{|V|})$ Farben färben.

2 Wahrscheinlichkeit Theorie

Def 2.1. Ein diskreter Wahrscheinlichkeitsraum ist bestimmt durch eine Ergebnismenge $\Omega = \{\omega_1, \omega_2, \ldots\}$ von Elementarereignissen. Jedem Elementarereignis ω_i ist eine Wahrscheinlichkeit $\Pr[\omega_i]$ zugeordnet, wobei wir fordern, dass $0 \leq \Pr[\omega_i] \leq 1$ und $\sum_{\omega \in \Omega} \Pr[\omega] = 1$. Eine Menge $E \subseteq \Omega$ heisst Ergeinis. Die Wahrscheinlichkeit $\Pr[E]$ eines Ereginisses ist definiert durch $\Pr[E] := \sum_{\omega \in E} \Pr[\omega]$. Ist E ein Ergeinis, so bezeichnen wir mit $\overline{E} := \Omega \setminus E$ das Komplementärereignis zu E.

Lemma 2.2. Für Ereignisse A, B gilt:

1. $\Pr[\varnothing] = 0, \Pr[\Omega] = 1$
2. $0 \leq \Pr[A] \leq 1$
3. $\Pr[\overline{A}] = 1 - \Pr[A]$
4. Wenn $A \subseteq B$, so folgt $\Pr[A] \leq \Pr[B]$

Satz 2.3 (Additionssatz)**.** Wenn $A_1, \ldots, A_n$ paarweise disjunkte Ereignisse sind, so gilt

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

Für eine unendliche Menge von disjunkten Ereignissen $A_1, A_2, \ldots$ gilt analog

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

Satz 2.5 (Siebformel)**.** Für Ereignisse $A_1, \ldots, A_n (n \geq 2)$ gilt:

$$\begin{aligned}\Pr\left[\bigcup_{i=1}^{n} A_i\right] &= \sum_{l=1}^{n} (-1)^{l+1} \sum_{1 \leq i_1 < \cdots < i_l \leq n} \Pr[A_{i_1} \cap \cdots \cap A_{i_l}] \\ &= \sum_{i=1}^{n} \Pr[A_i] - \sum_{i \leq i_1 < i_2 \leq n} \Pr[A_{i_1} \cap A_{i_2}] \\ &\quad + \sum_{1 \leq i_1 < i_2 < i_3 \leq n} \Pr[A_{i_1} \cap A_{i_2} \cap A_{i_3}] - \cdots \\ &\quad + (-1)^{n+1} \cdots \Pr[A_1 \cap \cdots \cap A_n]\end{aligned}$$

Cor 2.6 (Boolsche Ungleichung)**.** Für Ereignisse $A_1, \ldots, A_n$ gilt:

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

Analog gilt für eine unendliche Folge von Ereignissen $A_1, A_2, \ldots$, dass $\Pr[\bigcup_{i=1}^{\infty} A_i] \leq \sum_{i=1}^{\infty} \Pr[A_i]$.

Def 2.8. A und B seien Ereignisse mit $\Pr[B] > 0$. Die bedingte Wahrscheinlichkeit $\Pr[A|B]$ von A gegeben B ist definiert durch

$$\Pr[A|B] := \frac{\Pr[A \cap B]}{\Pr[B]}$$

Satz 2.10 (Multiplikationssatz)**.** Seien die Ereignisse $A_1, \ldots, A_n$ gegeben. Falls $\Pr[A_1 \cap \cdots \cap A_n] > 0$ ist, gilt

$$\begin{aligned}&Pr[A_1 \cap \cdots \cap A_n] = \\ &\qquad \Pr[A_1] \cdot \Pr[A_2|A_1] \cdots \Pr \Pr[A_n|A_1 \cap \cdots \cap A_{n-1}]\end{aligned}$$

Satz 2.13 (Totale Wahrscheinlichkeit)**.** Die Ereignisse $A_1, \ldots, A_n$ seien paarweise diskunkt und es gelte $B \subseteq A_1 \cup \ldots \cup A_n$. Dann folgt

$$\Pr[B] = \sum_{i=1}^{n} \Pr[B|A_i] \cdot \Pr[A_i]$$

Analog gilt für paarweise disjunkte Ereignisse $A_1, A_2, \ldots$ mit $B \subseteq \bigcup_{i=1}^{\infty} A_i$, dass

$$\Pr[B] = \sum_{i=1}^{\infty} \Pr[B|A_i] \cdot \Pr[A_i]$$

Satz 2.15 (Bayes)**.** Die Ereignisse $A_1, \ldots, A_n$ seien paarweise disjunkt. Ferner sei $B \subseteq A_1 \cup \ldots \cup A_n$ ein Ereignis mit $\Pr[B] > 0$. Dann gilt für ein beliebiges $i = 1, \ldots, n$

$$\Pr[A_i|B] = \frac{\Pr[A_i \cap B]}{\Pr[B]} = \frac{Pr[B|A_i] \cdot \Pr[A_i]}{\sum_{j=1}^{n} \Pr[B|A_j] \cdot \Pr[A_j]}$$

Analog gilt für paarweise disjunkte Ereignisse $A_1, A_2, \ldots$ mit $B \subseteq \bigcup_{i=1}^{\infty} A_i$, dass

$$\Pr[A_i|B] = \frac{\Pr[A_i \cap B]}{\Pr[B]} = \frac{Pr[B|A_i] \cdot \Pr[A_i]}{\sum_{j=1}^{\infty} \Pr[B|A_j] \cdot \Pr[A_j]}$$

Def 2.18. Die Ereignisse A und B heissen unabhängig, wenn gilt $\Pr[A \cap B] = Pr[A] \cdot \Pr[B]$

Def 2.22. Die Ereignisse $A_1, \ldots, A_n$ heissen unabhängig, wenn für alle Teilmengen $I \subseteq \{1, \ldots, n\}$ mit $I = \{i_1, \ldots, i_k\}$ gilt, dass

$$\Pr[A_{i_1} \cap \cdots \cap A_{i_k}] = \Pr[A_{i_1}] \cdots \Pr[A_{i_k}]$$

Eine unendliche Familie von Ereignissen A_i mit $i \in \mathbb{N}$ heisst unabhängig, wenn die Gleichung für jede endliche Teilmenge $I \subseteq \mathbb{N}$ erfüllt ist.

Lemma 2.23. Die Ereignisse $A_1, \ldots, A_n$ sind genau dann unabhängig, wenn für alle $(s_1, \ldots, s_n) \in \{0, 1\}^n$ gilt, dass

$$\Pr[A_1^{s_1} \cap \cdots \cap A_n^{s_n}] = \Pr[A_1^{s_1}] \cdots \Pr[A_n^{s_n}]$$

wobei $A_i^0 = \overline{A}_i$ und $A_i^1 = A_i$.

Lemma 2.24. Seien A, B und C unabhängige Ereignisse Dann sind auch $A \cap B$ und C bzw. $A \cup B$ und C unabhängig.

Def 2.25. Eine Zufallsvariable ist eine Abbildung $X : \Omega \to \mathbb{R}$, wobei Ω die Ergebnismenge eines Wahrscheinlichkeitsraum ist.

Dichtefunktion.

$$f_X : \mathbb{R} \to [0, 1], \quad x \mapsto \Pr[X = x]$$

Verteilungsfunktion.

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

Def 2.27. Zu einer Zufallsvariable X definieren wir den Erwartungswert $\mathbb{E}[X]$ durch

$$\mathbb{E}[X] := \sum_{x \in W_X} x \cdot \Pr[X = x]$$

sofern die Summe absolut konvergiert. Ansonsten sagen wir, dass der Erwartungswert undefiniert ist.

Lemma 2.29. Ist X eine Zufallsvariable, so gilt:

$$\mathbb{E}[X] = \sum_{\omega \in \Omega} X(\omega) \cdot \Pr[\omega]$$

Satz 2.30. Sei X eine Zufallsvariable mit $W_X \subseteq \mathbb{N}_0$. Dann gilt

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

Satz 2.32. Sei X eine Zufallsvariable. Für paarweise disjunkte Ereignisse $A_1, \ldots, A_n$ mit $A_1 \cup \cdots A_n = \Omega$ und $\Pr[A_1], \ldots, \Pr[A_n] > 0$ gilt

$$\mathbb{E}[X] = \sum_{i=1}^{n} \mathbb{E}[X|A_i] \cdot \Pr[A_i]$$

Für paarweise disjunkte Ereignisse $A_1, A_2, \ldots$ mit $\bigcup_{i=1}^{\infty} A_k = \Omega$ und $\Pr[A_1], \Pr[A_2], \ldots > 0$ gilt analog

$$\mathbb{E}[X] = \sum_{i=1}^{\infty} \mathbb{E}[X|A_i] \cdot \Pr[A_i]$$

Satz 2.33 (Linearität des Erwartungswerts)**.** Für Zufallsvariable $X_1, \ldots, X_n$ und $X := a_1 X_1 + \ldots + a_n X_n + b$ mit $a_1, \ldots, a_n, b \in \mathbb{R}$ gilt

$$\mathbb{E}[X] = a_1 \mathbb{E}[X_1] + \ldots + a_n \mathbb{E}[X_n] + b$$

Def 2.35 (Indikatorvariable)**.** Für ein Ereignis $A \subseteq \Omega$ ist die zugehörige Indikatorvariable X_A definiert durch:

$$X_A(\omega) := \begin{cases} 1, & \text{falls } \omega \in A \\ 0, & \text{sonst} \end{cases}$$

Für den Erwartungswert von X_A gilt: $\mathbb{E}[X_A] = \Pr[A]$.

Def 2.39. Für eine Zufallsvariable X mit $\mu = \mathbb{E}[X]$ definieren wir die Varianz $\mathrm{Var}[X]$ durch:

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

Die Grösse $\sigma := \sqrt{\mathrm{Var}[X]}$ heisst Standardabweichung von X.

Satz 2.40. Für eine beliebige Zufallsvariable X gilt

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

Satz 2.41. Für eine beliebige Zufallsvariable X und $a, b \in \mathbb{R}$ gilt

$$\mathrm{Var}[a \cdot X + b] = a^2 \cdot \mathrm{Var}[X]$$

2.1 Diskrete Verteilungen

Bmk (Bernoulli-Verteilung)**.**

$$X \sim \mathrm{Bernoulli}(p) \implies \mathbb{E}[X] = p \quad \mathrm{Var}[X] = p(1 - p)$$

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

Bmk (Binomial-Verteilung)**.**

$$X \sim \mathrm{Bin}(n, p) \implies \mathbb{E}[X] = np \quad \mathrm{Var}[X] = np(1 - p)$$

$$f_X(x) = \begin{cases} \binom{n}{x} p^x (1 - p)^{n-x} & x \in \{0, 1, \ldots, n\} \\ 0 & \text{sonst} \end{cases}$$

Bmk (Negativ Binomial-Verteilung)**.**

$$\mathbb{E}[Z] = \sum_{i=1}^{n} \mathbb{E}[X_i] = \frac{n}{p}$$

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

Bmk (Geometrisch-Verteilung)**.**

$$X \sim \mathrm{Geo}(p) \implies \mathbb{E}[X] = \frac{1}{p} \quad \mathrm{Var}[X] = \frac{1 - p}{p^2}$$

$$f_X(i) = \begin{cases} p(1 - p)^{i-1} & \text{für} i \in \mathbb{N} \\ 0 & \text{sonst} \end{cases}$$

Satz 2.45. Ist $X \sim \text{Geo}(p)$, so gilt für alle $s, t \in \mathbb{N}$:

$$\Pr[X \geq s + t \mid X > s] = \Pr[X \geq t]$$

Bmk (Poisson-Verteilung)**.**

$$X \sim \text{Po}(\lambda) \implies \mathbb{E}[X] = \text{Var}[X] = \lambda$$

$$f_X(i) = \begin{cases} \frac{e^{-\lambda}\lambda^i}{i!} & \text{für} i \in \mathbb{N} \\ 0 & \text{sonst} \end{cases}$$

2.2 Mehrere Zufallsvariablen

$\Pr[X = x, Y = y] = \Pr[\{\omega \in \Omega \mid X(\omega) = x, Y(\omega) = y\}]$

Bmk. Die gemeinsame Dichte von X und Y:

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

$$\implies$$

$$f_X(x) = \sum_{y \in W_Y} f_{X,Y}(x, y) \text{ bzw. } f_Y(y) = \sum_{x \in W_X} f_{X,Y}(x, y)$$

Def 2.52. Zufallsvariablen $X_1, \ldots, X_n$ heissen unabhängig, genau dann wenn für alle $(x_1, \ldots, x_n) \in W_{X_1} \times \ldots \times W_{X_n}$ gilt

$$\Pr[X_1 = x_1, \ldots, X_n = x_n] = \Pr[X_1 = x_1] \cdot \cdots \cdot \Pr[X_n = x_n]$$

Lemma 2.53. Sind $X_1, \ldots, X_n$ unabhängige Zufallsvariablen und $S_1, \ldots, S_n$ beliebige Mengen mit $S_i \subseteq W_{X_i}$, dann gilt

$$\Pr[X_1 \in S_1, \ldots, X_n \in S_n] = \Pr[X_1 \in S_1] \cdot \cdots \cdot \Pr[X_n \in S_n]$$

Cor 2.54. Sind $X_1, \ldots, X_n$ unabhängige Zufallsvariablen und ist $I = \{i + 1, \ldots, i_k\} \subseteq [n]$, dann sind $X_{i_1}, \ldots, X_{i_k}$ ebenfalls unabhängig.

Satz 2.55. Seien $f_1, \ldots, f_n$ reellwertige Funktionen ($f_i : \mathbb{R} \to \mathbb{R}$ für $i = 1, \ldots, n$). Wenn die Zufallsvariablen $X_1, \ldots, X_n$ unabhängig sind, dann gilt dies auch für $f_1(X_1), \ldots, f_n(X_n)$.

Satz 2.58. Für zwei unabhängige Zufallsvariablen X und Y und $Z := X + Y$. Es gilt

$$f_Z(z) = \sum_{x \in W_X} f_X(x) \cdot f_Y(z - x)$$

Satz 2.60 (Linearität des Erwartungswert)**.** Für Zufallsvariablen $X_1, \ldots, X_n$ und $X := a_1 X_1 + \cdots + a_n X_n$ mit $a_1, \ldots, a_n \in \mathbb{R}$ gilt

$$\mathbb{E}[X] = a_1 \mathbb{E}[X_1] + \cdots + a_n \mathbb{E}[X_n]$$

Satz 2.61 (Multiplikativität des Erwartungswerts)**.** Für unabhängige Zufallsvariablen $X_1, \ldots, X_n$ gilt

$$\mathbb{E}[X_1 \cdot \cdots X_n] = \mathbb{E}[X_1] \cdot \cdots \cdot \mathbb{E}[X_n]$$

Satz 2.62. Für unabhängige Zufallsvariablen $X_1, \ldots, X_n$ und $X := a_1 X_1 + \cdots + a_n X_n$ gilt

$$\text{Var}[X] = \text{Var}[X_1] + \ldots + \text{Var}[X_n]$$

Satz 2.60 (Waldsche Identität)**.** N und X seien zwei unabhängige Zufallsvariable, wobei für den Wertebereich von N gilt: $W_N \subseteq \mathbb{N}$. Weiter sei $Z := \sum_{i=1}^{N} X_i$ wobei $X_1, X_2, \ldots$ unabhängige Kopien von X seien. Dann gilt: $\mathbb{E}[Z] = \mathbb{E}[N] \cdot \mathbb{E}[X]$

Satz 2.67 (Ungleichung von Markov)**.** Sei X eine Zufallsvariable, die nur nicht-negative Werte annimmt. Dann gilt für alle $t \in \mathbb{R}$ mit $t > 0$, dass

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

Oder äquivalent: $\Pr[X \geq t \cdot \mathbb{E}[X]] \leq \frac{1}{t}$

Satz 2.68 (Ungleichung von Chebyshev)**.** Sei X eine Zufallsvariable und $t \in \mathbb{R}$ mit $t > 0$. Dann gilt

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

oder äquivalent: $\Pr[|X - \mathbb{E}[X]| \geq t\sqrt{\text{Var}[X]}] \leq \frac{1}{t^2}$

Satz 2.70 (Chernoff-Schranken)**.** Seien $X_1, \ldots, X_n$ unabhängig Bernoulliverteilte Zufallsvariablen mit $\Pr[X_i = 1] = p_1$ und $\Pr[X_1 = 0] = 1 - p_i$. Dann gilt für $X := \sum_{i=1}^{n} X_i$:

(i) $\Pr[X \geq (1 + \delta)\mathbb{E}[X]] \leq e^{-\frac{1}{3}\delta^2 \mathbb{E}[X]} \quad \forall 0 < \delta \leq 1$

(ii) $\Pr[X \leq (1 - \delta)\mathbb{E}[X]] \leq e^{-\frac{1}{2}\delta^2 \mathbb{E}[X]} \quad \forall 0 < \delta \leq 1$

(iii) $\Pr[X \geq t] \leq 2^{-t} \quad \text{für } t \geq 2e\mathbb{E}[X]$

2.9 Randomisierte Algorithmen

Satz 2.72. Sei A ein randomisierter Algorithmus, der nie eine falsche Antwort gibt, aber zuweilen '???' ausgibt, wobei

$$\Pr[A(I) \text{ korrekt}] \leq \epsilon$$

Dann gilt für alle $\delta > 0$: bezeichnet man mit A_δ den Algorithmus, der A solange aufruft bis entweder ein Wert verschieden von '???' ausgegeben wird (und A_δ diesen Wert dann ebenfalls ausgibt) oder bis $N = \epsilon^{-1} \ln \delta^{-1}$ mal '???' ausgegeben wurde (und A_δ dann ebenfalls '???' ausgibt), so gilt für den Algorithmus A_δ, dass

$$\Pr[A_\delta(I) \text{ korrekt}] \geq 1 - \delta$$

Satz 2.74 (Monte Carlo - Einseitiger Fehler)**.** Sei A ein randomisierter Algorithmus, der immer eine der beiden Antworten 'Ja' oder 'Nein' ausgibt, wobei

$$\Pr[A(I) = \text{Ja}] = 1 \text{ falls } I \text{ eine Ja-Instanz ist}$$

und

$$\Pr[A(I) = \text{Nein}] \geq \epsilon \text{ falls } I \text{ eine Nein-Instanz ist}$$

Dann gilt für alle $\delta > 0$: bezeichnet man mit $A_\delta(I)$ den Algorithmus, der A solange aufruft bis entweder der Wert 'Nein' ausgegeben wird (und A dann ebenfalls 'Nein' ausgibt) oder bis $N = \epsilon^{-1} \ln \delta^{-1}$ mal 'Ja' ausgegeben wurde (und A_δ dann ebenfalls 'Ja' ausgibt), so gilt für alle Instanzen I

$$\Pr[A_\delta(I) \text{ korrekt}] \geq 1 - \delta$$

Satz 2.75 (Monte Carlo - zweiseitiger Fehler)**.** Sei $\epsilon > 0$ und A ein randomisierter Algorithmus, der immer eine der beiden Antworten 'Ja' oder 'Nein' ausgibt, wobei

$$\Pr[A(I) \text{ korrekt}] \geq \frac{1}{2} + \epsilon$$

Dann gilt für alle $\delta > 0$: bezeichnet man mit A_δ den Algorithmus, der $N = 4\epsilon^{-2} \ln \delta^{-1}$ unabhängige Aufrufe von A macht und dann die Mehrheit der erhaltenen Antworten ausgibt, so gilt für den Algorithmus A_δ, dass

$$\Pr[A_\delta(I) \text{ korrekt}] \geq 1 - \delta$$

Satz 2.76. Sei $\epsilon > 0$ und A ein randomisierter Algorithmus für ein Maximierungsproblem, wobei gelte:

$$\Pr[A(I) \geq f(I)] \geq \epsilon$$

Dann gilt für alle $\delta > 0$ bezeichnet man mit A_δ den Algorithmus, der $N = \epsilon^{-1} \ln \delta^{-1}$ unabhängige Aufrufe von A macht und die beste der erhaltenen Antworten ausgibt, so gilt für den Algorithmus A_δ, dass

$$\Pr[A_\delta(I) \geq f(I)] \geq 1 - \delta$$

(Für Minimierungsprobleme gilt eine analoge Aussage wenn wir „$\geq f(I)$“ durch „$\leq f(I)$“ ersetzen.)

2.9.3 Primzahltest

Satz 2.77 (Kleiner fermatscher Satz)**.** Ist $n \in \mathbb{N}$ prim, so gilt für alle Zahlen $0 < a < n$

$$a^{n-1} \equiv 1 \mod n$$

Def (Carmichael-Zahl)**.** n heisst Carmichael-Zahl, falls n nicht prim ist und $PB_n = \mathbb{Z}_n^*$

$$PB_n := \{a \in [n-1] \mid \mathrm{ggT}(a, n) = 1 \land a^{n-1} \equiv_n 1\}$$

Bmk (Miller-Rabin-Primzahltest)**.**

1. $d, k \in \mathbb{N}$ mit $n - 1 = 2^k d$, d ungerade
2. Wähle $a \in [n-1]$, zufällig gleichverteilt
3. $a^d \not\equiv_n 1 \land \nexists i < k : a^{2^i d} \equiv_n n - 1 \implies$ nicht prim
4. ansonsten prim

Die Ausgabe 'nicht prim' ist immer richtig.
Die Ausgabe 'prim' ist falsch mit einer W'keit $\leq \frac{1}{4}$

2.9.4 Target Shooting

Satz 2.79. Seien $\delta, \epsilon > 0$. Falls $N \geq 3\frac{|U|}{|S|} \cdot \epsilon^{-2} \cdot \ln(\frac{2}{\delta})$, so ist die Ausgabe des Algorithmus TARGET-SHOOTING mit Wahrscheinlichkeit mindestens $1 - \delta$ im Intervall

$$\left[(1-\epsilon)\frac{|S|}{|U|}, (1+\epsilon)\frac{|S|}{|U|}\right]$$

(multiplikativer Fehler von $1 \pm \epsilon$)

2.9.5 Finden von Duplikaten

Bmk (Hashfunktion)**.** Hashfunktion $h : U \to [m]$ mit folgenden Eigenschaften:

- h ist effizient berechenbar
- h verhält sich wie eine Zufallsfunktion, d.h.

$$\forall u \in U \ \forall i \in [m] : \Pr[h(u) = i] = \frac{1}{m} \quad \text{unabhängig}$$

- $s_i = s_j \implies h(s_i) = h(s_j)$

Essenz: m viel kleiner als $|U|$ für Komprimierung.

Bmk (Kollisionen bei Hashing)**.** Kollisionen sind neue (unerwünschte) Duplikate im Hashmap. Sei $K_{i,j}$ die Bernoulli Variable mit

$$K_{i,j} = 1 \iff (i, j) \text{ is eine Kollision}$$

Es gilt

$$\Pr[K_{i,j} = 1] = \begin{cases} 1/m & \text{if } s_i \neq s_j, \\ 0 & \text{else} \end{cases} \implies \mathbb{E}[K_{i,j}] \leq \frac{1}{m}$$

$$\mathbb{E}[\#\text{Kollisionen}] = \sum_{1 \leq i < j \leq n} \mathbb{E}[K_{i,j}] \leq \binom{n}{2}\frac{1}{m}$$

Mit $m = n^2$ is der Mehraufwand durch Kollisionen konstant.
Laufzeit:

$$\mathcal{O}(n) + \mathcal{O}(n \log n) + \mathcal{O}(n + |\mathrm{Dupl}(S)|)$$

Bmk (Bloom-Filter)**.** Wähle k Hashfunktionen mit $h_i : U \to [m]$, $i = 1, \ldots, k$. Für jedes $s_i \in S$ haben wir einen Hashvektor $(x_1, \ldots, x_k) := (h_1(s_i), \ldots, h_k(s_i))$. Wir bereiten ein boolsches Feld $M[1..m]$ vor, anfangs alle Einträge auf 0.

Wir arbeiten uns durch die Elemente von $s \in S$, wobei wir für jedes x_i von s im Hashvektor je Element den Eintrag $M[x_i] = 1$ setzten. Falls schon alle $M[x_i]$ auf 1, gesetzt ist, fügen wir s in eine List $\mathcal{L}$ hinzu. Zum Schluss kontrolliert man nur diese Elemente von $\mathcal{L}$ nach Duplikaten.

$$X_i = 1 \iff \begin{cases} s_i \text{ tritt in } (s_1, \ldots, s_{i-1}) \text{ nicht auf und} \\ M[x_1] = \ldots = M[x_k] = 1 \text{ vor } s_i. \end{cases}$$

$$\mathbb{E}[\#\text{Fehler } \mathcal{L}] = \sum_{i=1}^{n} \mathbb{E}[X_i] \leq n \cdot \left(1 - \left(1 - \frac{1}{m}\right)^{k(n-1)}\right)^k$$

$$\begin{aligned} k \text{ und } m \text{ gross} &\implies \#\text{Falsche Einträge klein} \\ k \text{ gross} &\implies \text{langsamer} \\ m \text{ gross} &\implies \text{mehr Speicher} \end{aligned}$$

3 Algorithmen - Highlights

3.1 Graphen Algorithmen

3.1.1 Lange Pfade

Satz 3.1. Falls wir LONG-PATH für Graphen mit n Knoten in $t(n)$ Zeit entscheiden können, dann können wir in $t(2n-2) + \mathcal{O}(n^2)$ Zeit entscheiden, ob ein Graph mit n Knoten einen Hamilton Kreis hat.

Satz 3.2. Sei G ein Graph mit einem Pfad der Länge $k-1$.

1. Eine zufällige Färbung mit k Farben erzeugt einen bunten Pfad der Länge $k - 1$ mit Wahrscheinlichkeit $p_{\text{Erfolg}} \geq e^{-k}$.
2. Bei wiederholten Färbungen mit k Farben ist der Erwartungswert der Anzahl Versuche bis man einen bunten Pfad der Länge $k - 1$ erhält $\frac{1}{p_{\text{Erfolg}}} \leq e^k$.

Satz 3.3.

1. Der Algorithmus hat eine Laufzeit von $\mathcal{O}(\lambda(2e)^k km)$.
2. Antwortet der Algorithmus mit JA, dann hat der Graph einen Pfad der Länge $k - 1$.
3. Hat der Graph einen Pfad der Länge $k-1$, dann ist die Wahrscheinlichkeit, dass der Algortihmus mit NEIN antwortet, höchstens $e^{-\lambda}$.

3.1.2 Flüsse in Netzwerken

Def 3.4. Ein Netzwerk ist ein Tupel $N = (V, A, c, s, t)$, wobei gilt:

- (V, A) ist ein gerichteter Graph
- $s \in V$, source
- $t \in V \ \{s\}$, target
- $c : A \to \mathbb{R}_0^+$, capacity function

Def 3.5. Gegeben sei ein Netzwerk $N = (V, A, c, s, t)$. Ein Fluss in N ist eine Funktion $f : A \to \mathbb{R}$ mit den Bedingungen:

- $\forall e \in A : 0 \le f(e) \le c(e)$
- $\forall v \in V \ \{s, t\} :$

$$\sum_{u \in V:(u,v) \in A} f(u, v) = \sum_{u \in V:(u,v) \in A} f(v, u)$$

- $\text{val}(f) := \text{netoutflow}(s) :=$

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

Lemma 3.6. Der Nettozufluss der Senke gleicht dem Wert des Flusses, d.h. $\text{netinflow}(t) :=$

$$\sum_{u \in V:(u,v) \in A} f(u, t) - \sum_{u \in V:(t,u) \in A} f(t, u) = \text{val}(f)$$

Def 3.7. Ein s-t-Schnitt für ein Netzwerk $N = (V, A, c, s, t)$ ist eine Partition (S, T) von V (d.h. $S \cup T = V \wedge S \cap T = \emptyset$) mit $s \in S$ und $t \in T$. Die Kapazität eines s-t-Schnitts (S, T) ist definiert durch:

$$\text{cap}(S, T) := \sum_{(u,w) \in (S \times T) \cap A} c(u, w)$$

Lemma 3.8. Ist f ein Fluss und (S, T) ein s-t-Schnitt in einem Netzwerk $N = (V, A, c, s, t)$, so gilt

$$\text{val}(f) \le \text{cap}(S, T)$$

Satz 3.9 (Maxflow-Mincut)**.** Jedes Netzwerk $N = (V, A, c, s, t)$ erfüllt:

$$\max_{f \text{ Fluss in } N} \text{val}(f) = \min_{(S,T) \text{ s-t-Schnitt in } N} \text{cap}(S, T)$$

Def 3.10. Sei $N = (V, A, c, s, t)$ ein Netzwerk ohne entgegen gerichtete Kanten und sei f ein Fluss in N. Das Restnetzwerk $N_f := (V, A_f, r_f, s, t)$ ist wie folgt definiert:

1. Ist $e \in A$ mit $f(e) < c(e)$, dann ist e auch eine Kante in A_f mit $r_f(e) := c(e) - f(e)$.
2. ist $e \in A$ mit $f(e) > 0$, dann ist e^{opp} in A_f, mit $r_f(e^{\text{opp}}) = f(e)$.
3. Nur Kanten wie in 1. und 2. beschrieben finden sich in A_f.

$r_f(e), e \in A$ nennen wir die Restkapazität der Kante e.

Satz 3.11. Ein Fluss f in einem Netzwerk N ist ein maximaler Fluss $\iff$ es im Restnetzwerk N_f, keinen gerichteten s-t-Pad gibt. Für jeden solchen maximalen Fluss gibt es einen s-t-Schnitt (S, T) mit $\text{val}(f) = \text{cap}(S, T)$.

Satz 3.12. Sei $N = (V, A, c, s, t)$ ein Netzwerk mit $c : A \to \mathbb{N}_0^{\le U}, U \in \mathbb{N}$, ohne entgegen gerichtete Kanten. Dann gibt es einen ganzzahligen maximalen Fluss, der in Zeit $\mathcal{O}(mnU)$ berechnet werden kann.

Proposition 3.13 (Capacity-Scaling)**.** Sind in einem Netzwerk alle Kapazitäten ganzzahlig und höchstens U, so gibt es einen maximalen Fluss, der in Zeit $\mathcal{O}(mn(1 + \log U))$ berechnet werden kann (m Anzahl Kanten, n, Anzahl Knoten).

Proposition 3.14 (Dynamic Trees)**.** Der maximale Fluss eines Netzwerks kann in Zeit $\mathcal{O}(mn \log n)$ berechnet werden (m Anzahl Kanten, n Anzahl Knoten).

Lemma 3.15. Die maximale Grösse eines Matchings im bipartiten Graph G ist gleich dem Wert eines maximalen Flusses im Netzwerk N mit $c \equiv 1$ und

$$A := (\{s\} \times U) \cup \{(u, w) \in U \times W \mid \{u, w\} \in E\} \cup (W \times \{t\})$$

Bmk.

$$\begin{aligned} G = (V, E), u, v \in V &\mapsto N_G^* = (V, A, c, u, v) \\ A := &\{(x, y), (y, x) \mid \{x, y\} \in E\} \\ c &\equiv 1 \end{aligned}$$

$$\max_{f \text{ Fluss in } N} \text{val}(f) = \# \text{ intern knotendisjunkter } u\text{-}v\text{-Pfade}$$

Bmk (Bildsegmentierung)**.** Ein Bild ist ein Graph (P, E) mit Farbinformation $\chi : P \to$ Farben.

$\alpha : P \to \mathbb{R}_0^+$ $\quad \alpha_p$ grösser $\implies$ eher im Vordergrund

$\beta : P \to \mathbb{R}_0^+$ $\quad \beta_p$ grösser $\implies$ eher im Hintergrund

$\gamma : P \to \mathbb{R}_0^+$ $\quad \gamma_p$ grösser $\implies$ eher im gleichen Teil

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

zu maximieren ist äquivalent zur Minimierung von

$$q'(A, B) := \sum_{p \in A} \beta_p + \sum_{p \in B} \alpha_p + \sum_{e \in E}, \lambda_e$$

Die Problemstellung kann man durch ein Maxflow Problem im folgenden Netzwerk lösen: $N := (P \cup \{s, st\}, \vec{E}, c, s, t)$

- Neue Knoten s und t, Quelle und Senke im Netzwerk.
- $\forall p \in P : (s, p) \in \vec{E} \wedge \text{cap}((s, p)) = \alpha_p$
- $\forall p \in P : (p, t) \in \vec{E} \wedge \text{cap}((p, t)) = \beta_p$
- $\forall (p, p') \in E, p \ne p' : (p, p') \in \vec{E} \wedge (p', p) \in \vec{E} \wedge \text{cap}((p, p')) = \text{cap}((p', p)) = \lambda_e$

3.1.3 Minimale Schnitte in Graphen

Es werden ungerichtete Multigraphen für dieses Kapitel betrachtet. $\mu(G) \overset{\text{def}}{\Leftrightarrow}$ Kardinalität eines kleinsten Kantenschnitts in

Lemma 3.20. Sei G ein Graph und e eine Kante in G. Dann gilt $\mu(G \backslash e) \ge \mu(G)$ und falls es in G einen minimalen Schnitt C mit $e \notin C$ gibt, dann gilt $\mu(G \setminus e) = \mu(G)$.

Lemma 3.21. Sei $G = (V, E), n := |V|$. Für e gleichverteilt zufällig in E gilt:

$$\Pr[\mu(G) = \mu(G \setminus e)] \ge 1 - \frac{2}{n}$$

Bmk. $\hat{p}(G) :=$ W'keit, dass $\textsc{Cut}(\text{G})$ $\mu(G)$ ausgibt

Lemma 3.22. Es gilt für alle $n \ge 3$

$$\hat{p}(n) \ge (1 - \frac{2}{n}) \cdot \hat{p}(n - 1)$$

Lemma 3.23. Für alle $n \ge 2$ gilt $\hat{p}(n) \ge 1/\binom{n}{2}$. Das heisst, der Erwartungswert der #Wiederholungen, bis wir das erste Mal $\mu(G)$ ausgeben ist höchstens $\binom{n}{2}$

Def 3.24. Für den Algorithmus der $\lambda\binom{n}{2}$-maligen Wiederholung von $\textsc{Cut}(G)$ gilt:

1. Der Algorithmus hat eine Laufzeit von $\mathcal{O}(\lambda n^4)$.

2. Der kleinste angetroffene Wert ist mit einer W'keit von mindestens $1 - e^{-\lambda}$ gleich $\mu(G)$

Bmk (Bootstrapping)**.** Wir wenden $\textsc{Cut}(G)$ an, brechen nach bei G' mit t Knoten ab und wenden den Algorithmus $\textsc{Cut}(G')$ an.

$$\mathcal{O}(\lambda\left(\frac{n^4}{t^2} + n^2t^2\right)) \overset{t=\sqrt{n}}{=} \mathcal{O}(\lambda n^3)$$

Es bietet sich an, die gleiche Methode nun mit dem neuen $\mathcal{O}(n^3)$ Algorithmus statt dem $\mathcal{O}(n^4)$ Algorithmus zu versuchen und tatsächlich bekommen wir einen noch bessern, etc.
Im Limit entwickelt sich ein $\mathcal{O}(n^2 \text{polylog}(n))$-Algorithmus.

3.2 Geometrische Algorithmen

3.2.1 Kleinster umschliessender Kreis

Lemma 3.25. Für jede (endliche) Punktemenge $P \subset \mathbb{R}^2$ gibt es einen eindeutigen kleinsten umschliessenden Kreis $C(P)$.

Lemma 3.26. Für jede (endliche) Punktemenge $P \subset \mathbb{R}^2$ mit $|P| \geq 3$ gibt es eine Teilmenge $Q \subseteq P$, so dass $|Q| = 3$ und $C(Q) = C(P)$.

Lemma 3.28 (Sampling-Lemma)**.** Sei P' eine Menge von n (nicht unbedingt verschiedenen) Punkten und für $r \in \mathbb{N}$, für R zufällig gleichverteilt aus $\binom{P}{r}$. Dann ist die erwartete Anzahl Punkte von P, die ausserhalb von $C(R)$ liegen, höchstens $3\frac{n-r}{r+1} \leq 3\frac{n}{r+1}$.

Satz 3.29. Algorithmus $\textsc{Ranomized\_CleverVersion}$ berechnet den kleinsten umschliessenden Kreis von P in erwarteter Laufzeit $\mathcal{O}(n \log n)$.

3.2.2 Konvexe Hülle

Def 3.33.

- Liniensegment: $\overline{v_0v_1} := \{(1-\lambda)v_0\lambda v_1 \mid \lambda \in [0,1]\}$
- Konvexe Menge $C \subset \mathbb{R}^d$. $\forall v_0, v_1 \in C : \overline{v_0v_1} \subseteq C$
- Konvexe Hülle $\text{conv}(S) := \bigcap_{S \subseteq C \subseteq R^s, C \text{ konvex}} C$

$\text{conv}(P)$ wird durch ein Polygon P bestimmt, welches die Ecken Punkte aus P sind.

Lemma 3.34. $(q_0, q_1, \ldots, q_{h-1})$ ist die Eckfolge des $\text{conv}(P)$ umschliessenden Polygons gegen den Uhrzeigersinn genau dann wenn alle Paare (q_{i-1}, q_i), $i = 1, 2, \ldots h$ Randkanten von P sind.

Lemma 3.35. Seien $p = (p_x, p_y)$, $q = (q_x, q_y)$ und $r = (r_x, r_y)$ Punkte in R^2. Es gilt $q \neq r$ und p liegt links von qr genau dann wenn:

$$\det(p,q,r) := \begin{vmatrix} p_x & p_y & 1 \\ q_q & q_y & 1 \\ r_x & r_y & 1 \end{vmatrix} = \begin{vmatrix} q_x - p_x & q_y - p_y \\ r_x - p_x & r_y - p_y \end{vmatrix} > 0$$
$$\iff (q_x - p_x)(r_y - p_y) > (q_y - p_y)(r_x - p_x)$$

Lemma 3.36. Ist q eine Ecke der konvexen Hülle von P, so ist die Relation $\prec_q$ eine totale Ordnung auf $P \setminus \{q\}$. Für das Minimum p_min dieser Ordnung gilt, dass $qp_{\min}$ eine Randkante ist.

$$p_1 \prec_q p_2 :\iff p_1 \text{ rechts von } qp_2$$

Satz 3.37. Gegeben eine Menge P von n Punkten in allgemeiner Lage in R^2, berechnet der Algorithmus $\textsc{Jarvis-Wrap}$ die konvexe Hülle in Zeit $\mathcal{O}(nh)$, wobei h die Anzahl der Ecken der konvexen Hülle von P ist.

Bmk (Kollinearitäten)**.**

- Anfangspunkt q_0 als den Punkt mit der lexikographisch kleinster Koordinate.
- p rechts von qq_{next} muss ersetzt werden durch: p rechts von qq_{next} oder p auf der Geraden durch qq_{next} und $|qp| > |qq_{\text{next}}|$.
- Man kann nicht annehmen, dass die Punkte verschieden sind.

Bmk (Lokales Verbessern)**.** Sortiere P aufsteigend nach x-Koordinate: $(p_1, p_2, \ldots, p_n)$ und betrachte das Polygon $(p_1, p_2, \ldots, p_{n-1}, p_n, p_{n-1}, \ldots p_2)$.

Invarianten:

1. Der Teilpolygonzug $(p_1, \ldots, p_n)$ ist x-monoton und hat keinen Punkt in P unter sich.
2. Der Teilpolygonzug $(p_n, \ldots, p_1)$ ist x-monoton und hat keinen PUnkt in P über sich.
3. Der Teilpolygonzug $(p_1, \ldots, p_n)$ liegt nirgends über dem Teilpolygonzug $(p_n, \ldots, p_1)$

Satz 3.38. Gegeben eine Folge $p_1, p_2, \ldots, p_n$ nach x-Koordinate sortierter Punkte in allgemeiner Lage in R^2, berechnet der Algorithmus $\textsc{LocalRepair}$ die konvexe Hülle von $\{p_1, p_2, \ldots, p_n\}$ in Zeit $\mathcal{O}(n)$.