AnW – Woche 9

25.09.24
Nicolas Wehrli

Heute:

I. Kurze Kommentare zur Theorieaufgabe
II. Theory Recap
- Kleinster umschliessender Kreis
- Lange Pfade
- ~~Flüsse~~
- ~~Maxflow-Mincut~~ Theorem
- ~~Ford-Fulkerson~~

– – – – –

Wiederholung? Triangles? Prüfungsaufgabe?

~~III. Aufgabe~~
IV. Kahoot

I. Kurze Kommentare vom letzten Jahr

Generell sehr gut gelöst.

Definiert die Zufallsvariablen, die ihr verwendet!

Rechnet nicht einfach sondern schreibt hin was ihr macht!
Verhindert einfache Fehlen.

Bsp:

(d) Nimm an, dass $k = 1000 \log_2 n$ und $n \geq 2$. Zeige, dass mit Wahrscheinlichkeit mindesten 0.99 alle Spaziergänge ein Vergnügen sein werden.

Sei X = "Anzahl Strassen mit Blumen".

$$\Pr\left[X \geq \mathbb{E}[X] + \frac{k}{4}\right] \leq \dots$$

$$\vdots$$

$$\leq n^{-40}$$

1. Abschätzung pro Spaziergang

Sei $Z = Z_1 + \dots + Z_n$ die Anzahl Spaziergänge die kein Vergnügen sind. Wobei $Z_i \sim \mathrm{Ber}(p)$ mit $p \leq n^{-40}$

$$\mathbb{E}[Z] = n \cdot n^{-40} = n^{-39}$$ ← noch nicht das Resultat!

$$\Pr[Z \geq 1] \overset{\text{Markov}}{\leq} \frac{\mathbb{E}[Z]}{1} = n^{-39} \leq 2^{-39} < 0.01$$

$$\Rightarrow 1 - \Pr[Z \geq 1] \geq \underline{\underline{0.99}}$$

II. Theory Recap

Kleinster umschliessender Kreis

Problem: Gegeben eine endliche Punktemenge $P \subseteq \mathbb{R}^2$. Bestimme den kleinsten Kreis, der P umschliesst.

Für einen Kreis C, bezeichnen wir mit

$C^\bullet$ die geschlossene von C berandete Kreisscheibe.

Wir sagen **C umschliesst P**, falls $P \subseteq C^\bullet$. Punkte in P dürfen also auf C liegen.

Existenz nicht gezeigt

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

Beweis (nur für Eindeutigkeit). Angenommen, P hat zwei verschiedene kleinste umschliessende Kreise C_1 und C_2, beide mit Radius r und Mittelpunkten z_1 und z_2, $z_1 \neq z_2$. haben. Es gilt

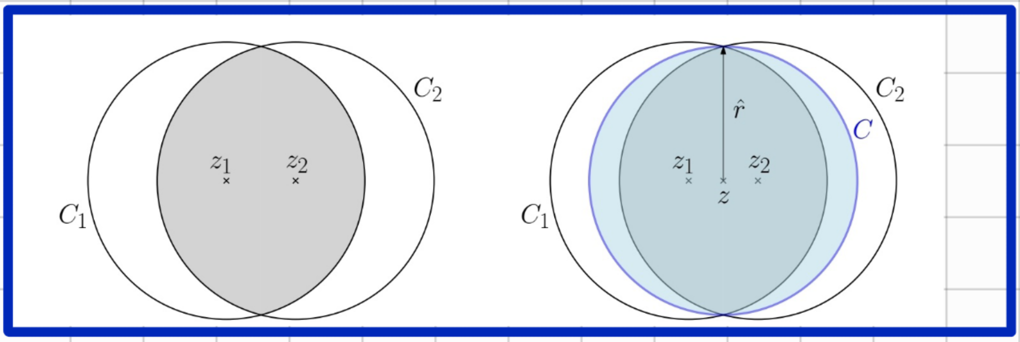
$$\boxed{P \subseteq C_1^\bullet \cap C_2^\bullet} .$$

Sei C der Kreis mit Mittelpunkt $z = \frac{1}{2}(z_1 + z_2)$ (Mittelpunkt des Liniensegments zwischen z_1 und z_2). Sein Radius $\hat{r}$ sei der Abstand von z zu den beiden Schnittpunkten von C_1 und C_2. Dann gilt

$$P \subseteq C_1^\bullet \cap C_2^\bullet \subseteq C^\bullet \text{ und } \hat{r} = \sqrt{r^2 - \left(\frac{|z_1 z_2|}{2}\right)^2}$$

($|z_1 z_2|$ Abstand von z_1 zu z_2).

Da $\hat{r} < r$, sind C_1 und C_2 nicht kleinste umschl. Kreise. □



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

Beweis ausgelassen siehe Skript.

Einfache Algorithmen – deterministisch

CompleteEnumeration(P)

for all $Q \in \binom{P}{3}$ **do** — $O(n^3)$ viele Teilmengen
  bestimme $C(Q)$
  if $P \subseteq C^\bullet(Q)$ **then** — $O(n)$ für den Test
    return $C(Q)$

Wir durchlaufen $\binom{n}{3}$ Mengen Q, berechnen $C(Q)$ in $O(1)$ Zeit, und prüfen $P \subseteq C^\bullet(Q)$ in $O(n)$ Zeit. Dies ergibt eine Laufzeit $O(n^4)$.

CompleteEnumerationSmart(P)

$r \leftarrow 0$
for all $Q \in \binom{P}{3}$ **do** bestimme $C(Q)$ $\longrightarrow$ Laufzeit $O(n^3)$.
  if radius$(C(Q)) > r$ **then** $O(1)$
    $C^* \leftarrow C(Q)$; $r \leftarrow$ radius$(C(Q))$ $O(1)$
return C^*

Nun versuchen wir einen randomisierten Algorithmus.

Randomised_PrimitiveVersion(P)

repeat forever
  wähle $Q \subseteq P$ mit $|Q| = 3$ zufällig und gleichverteilt
  bestimme $C(Q)$
  if $P \subseteq C^\bullet(Q)$ **then** — $O(n)$
    return $C(Q)$

Wir treffen auf das richtige Q mit Wahrscheinlichkeit $\geq 1/\binom{n}{3}$. Erwartete Anzahl der Versuche bis zum Erfolg ist $\leq \binom{n}{3}$, gibt erwartete Laufzeit $O(n^4)$.

Wir können 2 Dinge einfach verbessern.

1. Wir ziehen in jeder Iteration eine höhere (noch konstante) Anzahl Punkte. (Wir wählen $|Q| = 11$. Weshalb 11 sehen wir später.)

   Intuitiv erhöht sich die W'keit, dass die 3 kritischen Punkte von P in Q sind.

   Ändert die erwartete Laufzeit noch nicht asymptotisch.

2. Bei einer Iteration des Algorithmus lernen wir jeweils welche Punkte ausserhalb von $C(Q)$ liegen.

   $\Longrightarrow$ wichtiger für die Bestimmung von $C(P)$ als die Punkte in $C(Q)$.

verdopple in jeder Iteration der Punkte ausserhalb von $C(Q)$

Randomised_CleverVersion(P)

$P' \leftarrow P$
repeat
    wähle $Q \subseteq P'$ mit $\boxed{|Q| = 11}$ zufällig und gleichverteilt
    bestimme $C(Q)$
    if $P \subseteq C^\bullet(Q)$ **then return** $C(Q)$
    else verdopple alle Punkte von P' ausserhalb von $C(Q)$
forever

Realisierung der Verdoppelung?

Array num[] der Länge n für die n Punkte in P initialisiert mit 1.

$\Longrightarrow$ num[i] = "Häufigkeit von Punkt i"

$$\boxed{\text{Verdoppelung:} \quad num[i] \longleftarrow 2 \cdot num[i]}$$

Für die gleichverteilte Ziehung von $q_1, \ldots, q_m \in Q$ siehe Skript Lemma 3.27.

Ein Sampling Lemma für die Analyse

Lemma

Sei $r, N \in \mathbb{N}$, $r \leq N$ und $P' \subseteq \mathbb{R}^2$ eine Multimenge, $|P'| = N$. Für R zufällig gleichverteilt aus $\binom{P'}{r}$ gilt

$$\mathbb{E}(|\underbrace{P' \setminus C^\bullet(R)}_{\text{Punkte in } P' \text{ ausserhalb von } C(R)}|) \leq 3\frac{N-r}{r+1} \leq 3\frac{N}{r+1}$$

Beweis:

Wir definieren für den Beweis 2 Hilfsfunktionen, wobei $p \in P'$ und $R, Q \subseteq P'$:

$$\boxed{\operatorname{out}(p, R) := \begin{cases} 1 & \text{falls } p \notin C^\bullet(R) \\ 0 & \text{sonst} \end{cases}} \Longrightarrow \sum_{p \in P'} \operatorname{out}(p, R) = |P' \setminus C^\bullet(R)|$$

$$\boxed{\operatorname{essential}(p, Q) := \begin{cases} 1 & \text{falls } C(Q \setminus \{p\}) \neq C(Q) \\ 0 & \text{sonst.} \end{cases}} \Longrightarrow \sum_{p \in Q} \operatorname{ess}(p, Q) \leq 3$$

Für $p \notin R$ folgt leicht:

$$\mathrm{out}(p, R) = 1 \iff \mathrm{essential}(p, R \cup \{p\}) = 1.$$

$$\mathbb{E}\big[\underbrace{|P' \setminus C^*(R)|}_{\text{Zufallsvariable}}\big] = \sum_{R \in \binom{P'}{r}} |P' \setminus C^*(R)| \cdot p(R)$$

(Mappt jede Teilmenge R zu einer Zahl)

$$= \sum_{R \in \binom{P'}{r}} \Big(\sum_{s \in P'} \mathrm{out}(s, R) \Big) \cdot \frac{1}{\binom{N}{r}}$$

$$= \frac{1}{\binom{N}{r}} \cdot \sum_{R \in \binom{P'}{r}} \sum_{s \in P' \setminus R} \mathrm{out}(s, R)$$

$$= \frac{1}{\binom{N}{r}} \cdot \sum_{R \in \binom{P'}{r}} \sum_{s \in P' \setminus R} \mathrm{essential}(s, R \cup \{s\})$$

$$= \frac{1}{\binom{N}{r}} \sum_{Q \in \binom{P'}{r+1}} \underbrace{\sum_{p \in Q} \mathrm{essential}(p, Q)}_{\leq 3}$$

$$\leq \frac{1}{\binom{N}{r}} \sum_{Q \in \binom{P'}{r+1}} 3$$

$$= 3 \cdot \frac{\binom{N}{r+1}}{\binom{N}{r}} = 3 \, \frac{N - r}{r + 1}$$

$\boxtimes$

Dieses Lemma verwenden wir nun um die Laufzeit und Korrektheit zu beweisen.

Satz 3.29. Algorithmus RANDOMIZED_CLEVERVERSION berechnet den kleinsten umschliessenden Kreis von P in erwarteter Zeit $O(n \log n)$.

Beweis:

Wir definieren folgende Zufallsvariablen:
- Sei: T die Anzahl Iterationen des Algorithmus.
- Sei: X_k die Anzahl Punkte nach k Iterationen.

$\Rightarrow X_0 := n$

Zur Erinnerung:

Satz 2.32. Sei X eine Zufallsvariable. Für paarweise disjunkte Ereignisse $A_1, \ldots, A_n$ mit $A_1 \cup \cdots \cup A_n = \Omega$ 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

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

$$\Rightarrow \mathbb{E}[X_k] = \sum_{t=0}^{\infty} \mathbb{E}[X_k | \overbrace{X_{k-1} = t}^{A_t}] \cdot \Pr[X_{k-1} = t]$$

In der k-ten Runde, wenn $|P'| = t$ $(X_{k-1} = t)$ wird R zufällig gleichverteilt von $\binom{P'}{r}$ gezogen.

Es werden $|P' \setminus C^\circ(R)|$ Punkte verdoppelt. Wenn $X_{k-1} = t$ gegeben ist, folgt $X_k = |P' \setminus C^\circ(R)| + t$.

$$\Rightarrow \mathbb{E}[X_k | X_{k-1} = t] = \mathbb{E}[|P' \setminus C^\circ(R)|] + t$$

$$\leq \frac{3t}{r+1} + t \quad \left(\begin{array}{l}\text{per Sampling Lemma} \\ \text{mit } |P'| = t \; (= N)\end{array}\right)$$

$$\to \mathbb{E}[X_k] \leq \sum_{t=0}^{\infty} \left(1+\frac{3}{r+1}\right) t \cdot Pr[X_{k-1}=t]$$

$$= \left(1+\frac{3}{r+1}\right) \cdot \sum_{t=0}^{\infty} t \cdot Pr[X_{k-1}=t]$$

$$= \left(1+\frac{3}{r+1}\right) \cdot \mathbb{E}[X_{k-1}]$$

$$\Rightarrow \boxed{\begin{aligned} \mathbb{E}[X_k] &\leq \left(1+\frac{3}{r+1}\right)^k \cdot \mathbb{E}[X_0] \\ &= \left(1+\frac{3}{r+1}\right)^k \cdot n \end{aligned}}$$

obere Schranke
(1)

Da $C(P)$ maximal 3 kritische Punkte hat, war einer der Punkte mindestens $\frac{k}{3}$-Mal ausserhalb von $C(Q)$, wenn der Algorithmus nach k Runden noch nicht terminiert wurde.

Somit gibt es von mindestens von einem Punkt nach k Runden $2^{k/3}$ Kopien.

$$\to \mathbb{E}[X_k] = \underbrace{\mathbb{E}[X_k \mid T \geq k]}_{\geq 2^{k/3}} \cdot Pr[T \geq k] + \underbrace{\mathbb{E}[X_k \mid T < k] \cdot Pr[T<k]}_{\geq 0}$$

$$\geq 2^{k/3} \cdot Pr[T \geq k] + 0$$

$$\to \boxed{\mathbb{E}[X_k] \geq 2^{k/3} \cdot Pr[T \geq k]}$$

untere Schranke
(2)

$$2^{k/3} \cdot \Pr[T \geq k] \leq \mathbb{E}[X_k] \leq \left(1+\frac{3}{r+1}\right)^k n$$

$$\Longrightarrow \quad 2^{k/3} \cdot \Pr[T \geq k] \leq \left(1+\frac{3}{r+1}\right)^k \cdot n$$

$$\boxed{\Pr[T \geq k] \leq \left(\frac{1+\frac{3}{r+1}}{\sqrt[3]{2}}\right)^k \cdot n}$$

Nun haben wir eine Abschätzung für $\Pr[T \geq k]$ in Abhängigkeit von r (Grösse der Menge die wir ziehen).

Wir wollen nun, das unsere Abschätzung für $\Pr[T \geq k]$ mit zunehmendem k kleiner wird.

Also muss für r folgendes gelten:

$$\frac{1+\frac{3}{r+1}}{\sqrt[3]{2}} < 1 \qquad \Big| \cdot \left(2^{\frac{1}{3}}\right), -1$$

$$\frac{3}{r+1} < \sqrt[3]{2} - 1 \qquad \Big| \cdot (r+1) : \left(\sqrt[3]{2}-1\right)$$

$$\frac{3}{2^{\frac{1}{3}}-1} < r+1 \qquad \Big| -1$$

$$\frac{3}{2^{\frac{1}{3}}-1} - 1 \leq 10.55 < r$$

kleinstes r, dass diese Bedingung erfüllt, ist $\underline{\underline{r=11}}$.

Da $\Pr[T \geq k]$ eine W'keit ist, können wir sie wie folgt abschätzen:

$$\Pr[T \geq k] \leq \min(1, 0.995^k n) \quad \left(\text{wobei: } \left(\frac{1+\frac{3}{12}}{\sqrt[3]{2}}\right) \leq 0.995\right)$$

Mit all dem können wir nun $\mathbb{E}[T]$ abschätzen.

Zur Erinnerung:

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

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

Sei: $k_0 := \lceil -\log_{0.995}(n) \rceil$:

$$\begin{aligned}
\mathbb{E}[T] &= \sum_{k=1}^{\infty} \Pr[T \geq k] \\
&\leq \sum_{k=1}^{k_0} 1 + \sum_{k=k_0+1}^{\infty} 0.995^k n \\
&\overset{k=k_0+k'}{=} k_0 + \sum_{k'=1}^{\infty} 0.995^{k'} \cdot \underbrace{0.995^{k_0} \cdot n}_{\leq 1} \\
&\leq k_0 + \underbrace{\sum_{k'=1}^{\infty} 0.995^{k'}}_{\text{geometrische Reihe} \Rightarrow O(1)} \\
&= k_0 + O(1) \leq 200 \ln n + O(1)
\end{aligned}$$

Da jede Iteration in $O(n)$ läuft, ist die erwartete Laufzeit dieses Las Vegas Algorithmus $O(n \cdot \ln n)$.

$\boxtimes$

Lange Pfade

LONG-PATH Problem. Gegeben (G, B), G ein Graph und $B \in \mathbb{N}_0$, stelle fest ob es einen Pfad der Länge B in G gibt.

Ein **Pfad der Länge** ℓ in einem Graph $G = (V, E)$ ist eine Folge von paarweise verschiedenen Knoten

$$\langle v_0, v_1, \dots, v_\ell \rangle, \text{ mit } \{v_{i-1}, v_i\} \in E \text{ für } i = 1, \dots \ell.$$

Es liegen $\ell + 1$ Knoten auf einem Pfad der Länge ℓ.

Dieses Problem ist relativ schwer.

Generelle Version vom Hamiltonpfadproblem.

(Wir kennen (noch) keinen Polynomialzeit-Algorithmus)

In den meisten Fällen ist B relativ kurz $(B = O(\log n))$.

Für diesen Fall präsentieren wir eine Lösung.

Notation und Eigenschaften

- $[n] := \{1, \dots, n\}$

- $[n]^k = \{(a_1, a_2, \dots, a_k) \mid \forall i \in [k]: a_i \in [n]\}$ "alle Folgen der Länge k"
- $|[n]^k| = n^k$
- $\binom{[n]}{k} = \{S \subseteq [n] \mid |S| = k\}$, $\left|\binom{[n]}{k}\right| = \binom{n}{k}$
- Nachbarschaft von v in G: $N(v) = \{u \in V \mid \{u,v\} \in E\}$
- Handschlag-Lemma: Graph $G=(V,E)$: $\sum_{v \in V} \deg(v) = 2|E|$
- Für $c, n \in \mathbb{R}^+$: $c^{\log n} = n^{\log c}$ etc.
  (Auch $2^{O(\log n)} = n^{O(1)}$ ist polynomiell in n)
- Für $n \in \mathbb{N}_0$: $\sum_{i=0}^{n} \binom{n}{i} = 2^n$
  $\frac{n!}{n^n} \geq e^{-n}$ (folgt aus $e^n = \sum_{i=0}^{\infty} \frac{n^i}{n!} \geq \frac{n^n}{n!}$)

Zuerst ein anderes Problem:

Bunte Pfade

$k \in \mathbb{N}$. Graph $G = (V, E)$ mit Färbung $\gamma\colon V \to [k]$ (nicht notwendigerweise gültige Färbung!).

Ein Pfad heisst **bunt**, falls alle seine Knoten verschiedene Farben haben.

COLORFUL-PATH Problem. Gegeben (G, γ), $G = (V, E)$ ein Graph und $\gamma\colon V \to [k]$, stelle fest ob es einen bunten Pfad der Länge $k - 1$ (d.h. mit k Knoten) in G gibt.

Nun DP:

Für $v \in V$ und $i \in \mathbb{N}_0$: $DP[v][i] = P_i(v) =$

$$\left\{ S \in \binom{[k]}{i+1} \;\middle|\; \exists \text{ ein mit } S \text{ gefärbter bunter Pfad der in } v \text{ endet} \right\}$$

Die Farbe von v $\gamma(v)$ ist immer in $S \in P_i(v)$ enthalten.

$\Rightarrow \forall S \in P_i(v): \; \gamma(v) \in S$

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

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

$\exists$ bunter Pfad mit k Knoten $\Longleftrightarrow \bigcup_{v \in V} P_{k-1}(v) \neq \emptyset$

<u>DP Rekursion (Rückbezug auf Lösung kleinerer Fälle)</u>

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

Ein Iterationsschritt der äusseren Schleife.

(quasi die i-te Spalte von $DP[v][i]$ berechnen)

Berechnung aller $P_i(v)$, $v \in V$, mit den $P_{i-1}(v)$, $v \in V$, gegeben:

Bunt(G, i)	G ein γ-gefärbter Graph
for all $v \in V$ **do**	$\forall v \in V$
$\quad P_i(v) \leftarrow \emptyset$	
$\quad$ **for all** $x \in N(v)$ **do**	$\deg(v)$
$\quad\quad$ **for all** $R \in P_{i-1}(x)$ mit $\gamma(v) \overset{i}{\notin} R$ **do**	$\lvert P_{i-1}(v)\rvert \cdot i$
$\quad\quad\quad P_i(v) \leftarrow P_i(v) \cup \{R \cup \{\gamma(v)\}\}$	

Analyse:

Da $P_{i-1}(v) \subseteq \binom{[n]}{i}$ folgt $|P_{i-1}(v)| \leq \binom{n}{i}$

$$\Rightarrow O\Big(\underbrace{\sum_{v\in V} \deg(v)}_{\text{Handschlag: } 2m} \cdot \binom{k}{i} \cdot i\Big) = \underline{O\Big(\binom{n}{i} \cdot i \cdot m\Big)}$$

Regenbogen(G, γ)	G Graph, γ k-Färbung
for all $v \in V$ **do** $P_0(v) \leftarrow \{\{\gamma(v)\}\}$	
for $i = 1..k-1$ **do** Bunt(G, i)	
return $\bigcup_{v \in V} P_{k-1}(v) \neq \emptyset$	

$$O\Big(|V| + \sum_{i=1}^{k-1} \Big(\binom{k}{i} \cdot i \cdot m\Big) + |V|\Big) = O(2^k k m)$$

$$\Big(\sum_{i=1}^{k-1} \Big(\binom{k}{i} \cdot i \cdot m\Big) \leq \sum_{i=1}^{k-1} \binom{k}{i} \cdot k \cdot m \leq 2^k k m\Big)$$

D.h. für $k \leq \log n$: Laufzeit $O(mn \log n)$,
$k = O(\log n)$: Laufzeit $O(\operatorname{poly}(n))$.

Zurück zum Long-Path Problem.

Gibt es einen Pfad der Länge B in $G=(V,E)$.

Setze $k := B + 1$, färbe G zufällig mit k Farben, und suche einen bunten Pfad mit k Knoten.

Nehmen wir an es existiert ein Pfad P mit k Knoten.

Diese k Knoten des Pfades können auf k^k verschiedene Arten mit k Farben gefärbt werden.

Bei $k!$ dieser Färbungen, hat jeder Knoten eine andere Farbe.

$$p_{\text{Erfolg}} \geq \frac{k!}{k^k} \geq e^{-k}$$

Per Geometrischer Verteilung mit $\text{Geo}(e^{-k})$, braucht man im Erwartungswert e^k Versuche.

Angenommen G hat einen Pfad mit k Knoten.

Ein Versuch:

- Laufzeit $O(2^k km)$. $\quad p_{\text{Erfolg}} \geq e^{-k}$.

$\lceil \lambda e^k \rceil$ Versuche:

- Laufzeit $O(\lambda (2e)^k km)$.
- W'keit, dass der Algorithmus den Pfad nicht findet ist

$$\leq \left(1 - e^{-k}\right)^{\lceil \lambda e^k \rceil} \leq \left(e^{-e^{-k}}\right)^{\lceil \lambda e^k \rceil} \leq e^{-\lambda}.$$

- Will man einen Pfad der Länge B tatsächlich finden (statt nur die Existenz festzustellen), kann man den Algorithmus leicht adaptieren. Man merkt sich dazu einfach zu jedem $S \in P_i(v)$ <u>einen</u> genau mit S gefärbten bunten Pfad $\langle u_0, u_1, \ldots, u_i \rangle$, $u_i = v$.
- Das Problem wird einfach in gerichteten azyklischen Graphen: Man gewichtet alle Kanten einfach mit -1 und berechnet kürzeste Pfade (siehe *Algorithmen und Datenstruktur*-Vorlesung im Herbst).

Unötig, da wir den Beweis gerade gemacht haben.

Aufgabe 1 – *Kleinster Umschliessender Ball*

In dieser Aufgabe sollen Sie den Algorithmus zum kleinsten umschliessenden Kreis auf den dreidimensionalen Fall übertragen.

(a) Zeigen Sie die dreidimensionale Variante des *Sampling Lemma*:
Sei $P \subseteq \mathbb{R}^3$ eine Menge von n (nicht unbedingt verschiedenen) Punkten im dreidimensionalen Raum, $r \in \mathbb{N}$, und sei R zufällig gleichverteilt aus $\binom{P}{r}$. Sei X die Anzahl Punkte von P, die ausserhalb des kleinsten umschliessenden Balles $B(R)$ von R liegen. Dann ist $\mathbb{E}[X] \leq 4\frac{n-r}{r+1}$.

(b) Entwerfen Sie einen Algorithmus, der als Input eine Menge $P \subseteq \mathbb{R}^3$ von n Punkten im dreidimensionalen Raum bekommt, und der in erwarteter Zeit $O(n \log n)$ den kleinsten umschliessenden Ball $B(P)$ von P bestimmt.

Sie brauchen dabei nicht genau die Datenstrukturen zu spezifizieren, die Sie verwenden. Insbesondere dürfen Sie davon ausgehen, dass Sie für gegebene Zahlen $d_1, \ldots, d_n \in \mathbb{N}$ in Zeit $O(n)$ einen Index i mit Wahrscheinlichkeit proportional zu d_i ziehen können, also mit $\Pr[i] = \frac{d_i}{D}$, wobei $D = \sum_{i=1}^{n} d_i$ ist.

Hinweis: Sei $P \subseteq \mathbb{R}^3$ eine Menge von n (nicht unbedingt verschiedenen) Punkten im dreidimensionalen Raum. Sie dürfen für die Aufgabe die folgenden Fakten ohne weitere Begründung verwenden.

1. $B(P)$ ist eindeutig bestimmt.
2. Ist $Q \subseteq P$, so ist $\mathrm{Vol}(B(Q)) \leq \mathrm{Vol}(B(P))$.
3. Für jede endliche Menge $Q \subseteq \mathbb{R}^3$ gibt es eine Teilmenge $Q' \subseteq Q$ mit $|Q'| \leq 4$ sodass $B(Q') = B(Q)$.
   Insbesondere kann $essential(p, Q) = 1$ nur für höchstens vier Punkte $p \in Q$ erfüllt sein.

Hinweis zu (a): Gehen Sie wie im Beweis von Lemma 3.28 vor. Benutzen Sie dafür insbesondere die Grössen

$$out(p, R) := \begin{cases} 1 & \text{falls } p \notin B(R) \\ 0 & \text{sonst} \end{cases} \qquad \text{und} \qquad essential(p, Q) := \begin{cases} 1 & \text{falls } B(Q \setminus \{p\}) \neq B(Q) \\ 0 & \text{sonst.} \end{cases}$$

Lösung zu Aufgabe 1 – *Kleinster Umschliessender Ball*

(a) *Proof.* Für den Beweis definieren wir uns zwei Hilfsfunktionen. Für alle $p \in P$, $R, Q \subseteq P$ sei

$$out(p,R) := \begin{cases} 1 & \text{falls } p \notin B(R) \\ 0 & \text{sonst} \end{cases} \qquad \text{und} \qquad essential(p,Q) := \begin{cases} 1 & \text{falls } B(Q \setminus \{p\}) \neq B(Q) \\ 0 & \text{sonst.} \end{cases}$$

Man beachte, dass $\sum_{p \in P \setminus R} out(p,R)$ die Anzahl der Punkte ausserhalb von $B(R)$ ist.

Leicht überzeugt man sich davon, dass beide Funktionen für alle $p \in P \setminus R$ in folgender Beziehung stehen.

$$out(p,R) = 1 \qquad \Longleftrightarrow \qquad essential(p, R \cup \{p\}) = 1.$$

Da $essential(p,Q) = 1$ nur für höchstens vier Punkte $p \in Q$ erfüllt sein kann, erhalten wir für die Anzahl X der Punkte ausserhalb von $B(R)$:

$$\begin{aligned}
\mathbb{E}[X] &= \frac{1}{\binom{n}{r}} \sum_{R \in \binom{P}{r}} \sum_{s \in P \setminus R} out(s,R) \\
&= \frac{1}{\binom{n}{r}} \sum_{R \in \binom{P}{r}} \sum_{s \in P \setminus R} essential(s, R \cup \{s\}) \\
&= \frac{1}{\binom{n}{r}} \sum_{Q \in \binom{P}{r+1}} \underbrace{\sum_{p \in Q} essential(p,Q)}_{\leq 4} \\
&\leq \frac{1}{\binom{n}{r}} \sum_{Q \in \binom{P}{r+1}} 4 \;=\; 4 \cdot \frac{\binom{n}{r+1}}{\binom{n}{r}} \;=\; 4 \frac{n-r}{r+1},
\end{aligned}$$

□

(b) Die Idee des Algorithmus ist auch hier eine kleine Menge an Punkten zufällig auszuwählen und zu testen, ob dessen kleinste umschliessende Kugel die ganze Punktemenge P enthält. Falls dies nicht der Fall ist, verdoppelt man alle Punkte ausserhalb der Kugel und wiederholt diesen Schritt.
Für eine endliche Punktemenge $P \subset \mathbb{R}^3$ ergibt dies folgenden Algorithmus:

Algorithm 1 UMSCHLIESSENDER_BALL_ALGORITHMUS($\mathcal{P}$)

repeat forever
  wähle $R \subseteq P$ mit $|R| = 21$ zufällig und gleichverteilt
  bestimme $B(R)$
  if $P \subseteq B(R)$ **then**
    return $B(R)$
  verdoppele alle Punkte von P ausserhalb von $B(R)$

Die Korrektheit des Algorithmus ist nach Konstruktion klar. Wir zeigen nun dass dieser Algorithmus eine erwartete Laufzeit von $O(n \log(n))$ hat.
Bei einem einzelnen Durchlauf der Wiederholungsschlaufe müssen wir ein zufälliges R auswählen. Anstatt die Punkte wirklich zu verdoppeln können wir wie im Skript bei jedem Punkt $i \in P$ einen Index d_i hinzufügen, der besagt, wieviele Kopien dieses Punktes vorhanden sind. Dann können wir in Zeit $O(n)$ einen Punkt $i \in P$ mit Wahrscheinlichkeit $d_i / \sum_{i \in P} d_i$ ziehen. Die eindeutige kleinste umschliessende Kugel wird in $O(1)$ berechnet, da wir nur 21 Punkte betrachten. Um zu überprüfen ob $P \subset B(R)$gilt, durchlaufen wir alle Punkte $i \in P$ und testen,

ob $i \in B(R)$. Dies braucht ebenfalls Zeit $O(n)$. Das verdoppeln der Indizes der Punkte ausserhalb von $B(R)$ braucht Zeit $O(1)$ für jeden Punkt in $\mathcal{P} \setminus B(R)$. Insgesamt brauchen wir folglich Zeit $O(n)$ für jede Iteration des Algorithmus.

Analog zum zweidimensionalen Fall definieren wir nun die Zufallsvariable T als die Anzahl Iterationen des Algorithmus. Des weiteren sei X_k gleich der Anzahl Punkte nach k Iterationen. Da wir in der k-ten Iteration zufällig eine gleichverteilt zufällige Menge R von 21 Punkten aus einer Menge von X_{k-1} Punkten auswählen, folgt aus Aufgabe (a), dass in Erwartung weniger als $\frac{4}{22}X_{k-1}$ Punkte ausserhalb von $B(R)$ liegen. Somit erwarten wir, dass X_k kleiner als $(1+\frac{4}{22})X_{k-1}$ ist. Dies gibt uns eine obere Schranke für den Erwartungswert $\mathbb{E}[X_k]$.

$$
\begin{aligned}
\mathbb{E}[X_k] &= \sum_{t\geq n} \mathbb{E}[X_k \mid X_{k-1}=t] \cdot \Pr[X_{k-1}=t] \\
&\leq \sum_{t\geq n} (1+\frac{4}{22})t \cdot \Pr[X_{k-1}=t] \\
&= (1+\frac{2}{11}) \cdot \sum_{t\geq n} t \cdot \Pr[X_{k-1}=t] \\
&= (1+\frac{2}{11}) \cdot \mathbb{E}[X_{k-1}].
\end{aligned}
$$

Und per Induktion mit $X_0 = n$ gilt $\mathbb{E}[X_k] \leq (1+\frac{2}{11})^k \cdot n$.

Wir benutzen nun, dass es 4 Punkte in $\mathcal{P}$ gibt, welche die kleinste umschliessende Kugel eindeutig bestimmen. Nennen wir die Menge dieser Punkte Q_0, somit gilt $B(\mathcal{P}) = B(Q_0)$. Wählt unser Algorithmus eine Menge Q sodass dessen kleinste umschliessende Kugel $B(Q)$ die Menge Q_0 umschliesst, so ist $B(Q)$ mindestens so gross wie $B(Q_0)$. Gleichzeitig gilt aber, dass $B(Q)$ kleiner oder gleich $B(P)$ ist, da Q eine Teilmenge von $\mathcal{P}$ ist. Deshalb, da die kleinste umschliessende Kugel eindeutig bestimmt ist, gilt $B(Q) = B(Q_0) = B(\mathcal{P})$ und der Algorithmus terminiert.

Entsprechend muss in jeder Runde, in denen der Algorithmus nicht terminiert, mindestens einer der 4 Punkte von Q_0 ausserhalb von $B(Q)$ liegen und wird in dieser Runde verdoppelt. Falls der Algorithmus länger als k Runden läuft, gibt es somit mindestens einen Punkt der $k/4$ viele Runden ausserhalb der Kugel war und verdoppelt wurde. Also gibt es mindestens $2^{k/4}$ Kopien von diesem Punkt. Das bedeutet aber, dass der Erwartungswert von X_k, der Gesamtanzahl Punkten nach k Runden, mindestens $2^{k/4}$ ist, falls der Algorithmus nach k Iterationen noch nicht terminiert hat. Somit erhalten wir eine untere Schranke für den Erwartungswert $\mathbb{E}[X_k]$.

$$
\mathbb{E}[X_k] = \underbrace{\mathbb{E}[X_k \mid T \geq k]}_{\geq 2^{k/4}} \cdot \Pr[T \geq k] + \underbrace{\mathbb{E}[X_k \mid T < k]}_{\geq 0} \cdot \Pr[T < k] \geq 2^{k/4} \cdot \Pr[T \geq k].
$$

Zusammen mit der obere Schranke an $\mathbb{E}[X]$, welche wir oben hergeleitet haben, erhalten wir nun eine Abschätzung für $\Pr[T \geq k]$:

$$
\begin{aligned}
& 2^{k/4} \cdot \Pr[T \geq k] \leq \mathbb{E}[X_k] \leq (1+\frac{2}{11})^k \cdot n \\
\Rightarrow \qquad & \Pr[T \geq k] \leq \frac{(1+\frac{2}{11})^k \cdot n}{2^{k/4}} \leq \left(\frac{(1+\frac{2}{11})}{2^{1/4}}\right)^k \cdot n \leq (0.994)^k \cdot n
\end{aligned}
$$

Und dieser Wert fällt exponentiell mit k.

Die erwartete Anzahl Runden lässt sich jetzt gegen oben beschränken. Dafür benutzen wir

$\Pr[T \geq k] \leq \min\{1, 0.994^k n\}$ und $k_0 := \lceil -\log_{0.994} n \rceil$:

$$\begin{aligned}
\mathbb{E}[T] &= \sum_{k\geq 1} \Pr[T \geq k] \\
&\leq \sum_{k=1}^{k_0} 1 + \sum_{k>k_0} 0.994^k n \\
&= \underbrace{\sum_{k=1}^{k_0} 1}_{=k_0} + \sum_{k\geq k_0} 0.994^{k-k_0} \cdot \underbrace{0.994^{k_0} n}_{\leq 1} \\
&= k_0 + \sum_{k'\geq 1} 0.994^{k'} \\
&\leq \lceil 166.166 \cdot \log(n) \rceil + 166.667 = O(\log(n)).
\end{aligned}$$

Da wir in jeder Iteration des Algorithmus $O(n)$ viele Operationen brauchen, haben wir hiermit gezeigt, dass der Algorithmus eine erwartete Laufzeit von $O(n \log(n))$ hat.