Organisatorisches

nächste Woche: Mittwoch (28.05.) zusätzliche Übungsstunde im LEE C 104
14:15–16:00

→ Fokus auf wie für A&W lernen, zusätzliche Ressourcen,
Recap, Q&A

→ Bitte äussert eure Wünsche bis Samstag Abend

→ meine Vorschläge: · Beweisaufgabe im Prüfungsstil
· systematische Ansätze für Wahrscheinlichkeits-DP

Mittwoch (28.05.) 13–14 Uhr in HG F7

grosses Abschlusskahoot mit anderen Übungsgruppen (und Preisen)

Bootstrapping

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

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

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

Idee: Wiederhole den Teil, wo die Fehlerw'keit gross ist

Abbruch des Algorithmus nach t Knoten,
dann Wiederholen des Algorithmus für t Knoten in $O(t^4)$ und Erfolgsw'keit $\geq \frac{e-1}{e}$

$$\hat{p}_t(n) \geq \frac{n-2}{n} \cdot \frac{n-3}{n-1} \cdot \ldots \cdot \frac{t}{t+2} \cdot \frac{t-1}{t+1} \cdot \hat{p}_t(t) \geq \underbrace{\frac{t(t-1)}{n(n-1)} \cdot \frac{e-1}{e}}_{=:\overline{p_t}(n)}$$

$\lambda \cdot \frac{1}{\overline{p_t}(n)}$ Wiederholungen des ganzen Algorithmus:

Fehlerw'keit $\leq e^{-\lambda}$

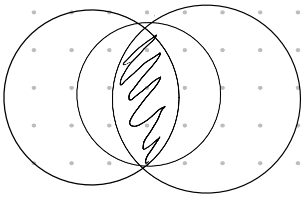
$$\text{Laufzeit: } O(\underbrace{\lambda \underbrace{\frac{n(n-1)e}{t(t-1)(e-1)}}_{\overline{p_t}(n)}}_{\#\text{ Wdh}} \cdot (\underbrace{n(n-t)}_{\substack{\text{Kontrahieren}\\ \text{bis } t \text{ Knoten}\\ \text{übrig}}} + t^4)) = O(\lambda(\frac{n^4}{t^2} + n^2t^2)) \overset{t=\sqrt{n}}{=} O(\lambda n^3)$$

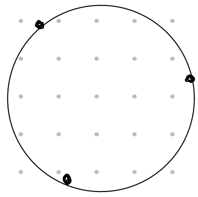
möglich bis zu $O(n^2 \text{polylog}(n))$ – Algorithmus

Kleinster umschliessender Kreis

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

Intuition/Beweis:





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

→ es liegen mind. 3 Punkte auf dem Rand

naiver Algorithmus:

CompleteEnumeration(P)

for all $Q \subseteq P$ mit $|Q| = 3$ **do**
  bestimme $C(Q)$
  if $P \subseteq C^{\bullet}(Q)$ **then** $\leftarrow O(\ )$
    return $C(Q)$

Wie viele potenzielle Mengen für Q gibt es?
$\binom{n}{3}$
Gesamtlaufzeit: $O(n^4)$

Was ist, wenn wir die Menge Q zufällig ziehen?
→ gleich Laufzeit im Erwartungswert

Idee: Nach jedem Schritt verdoppeln wir die Punkte ausserhalb $C(Q)$

Randomised_CleverVersion(P)

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

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

Beweis: $\text{out}(p,R) = \begin{cases} 1 & \text{falls } p \notin C(R) \\ 0 & \text{sonst} \end{cases}$

$$\text{essential}(p,Q) = \begin{cases} 1 & \text{falls } C(Q \setminus \{p\}) \neq C(Q) \\ 0 & \text{sonst} \end{cases}$$

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

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

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

$$= \frac{1}{\binom{n}{r}} \sum_{R \in \binom{P}{r}} \sum_{s \in P \setminus R} \text{essential}(s, \underbrace{R \cup \{s\}}_{Q})$$

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

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

$\square$

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

$X_k = \#$Punkte nach k Iterationen; $X_0 = n$

$$E[X_k] \leq \left(1 + \frac{3}{r+1}\right) E[X_{k-1}] \xrightarrow{\text{Induktion}} E[X_k] \leq \left(1 + \frac{3}{r+1}\right)^k \cdot n$$

obere Schranke: $E[X_k] = \sum_{t=0}^{\infty} E[X_k \mid X_{k-1} = t] \cdot Pr[X_{k-1} = t]$

$$\overset{\text{Lemma 3.28}}{\leq} \sum_{t=0}^{\infty} \left(1 + \frac{3}{r+1}\right) t \cdot Pr[X_{k-1} = t]$$

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

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

untere Schranke: $E[X_k] = E[X_k \mid T \geq k] \cdot Pr[T \geq k] + \underbrace{E[X_k \mid T < k]}_{\geq 0} \cdot Pr[T < k]$

$\geq 2^{\frac{k}{3}} \cdot Pr[T \geq k] \rightarrow$ wenn noch nicht terminiert muss immer einer der max. drei essentiellen Punkte in mind. $\frac{k}{3}$ Iterationen ausserhalb gelegen haben

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

$\rightarrow$ für $r = 11$ folgt $Pr[T \geq k] \leq \min\{1, 0.995^k n\}$

#Iterationen $\qquad k_0 = -\log_{0.995}(n)$

$$E[T] = \sum_{k \geq 1} Pr[T \geq k]$$

$$\leq \sum_{k=1}^{k_0} 1 + \sum_{k > k_0} Pr[T \geq k]$$

$$\overset{k = k_0 + k'}{=} k_0 + \sum_{k' \geq 1} 0.995^{k'} \cdot \underbrace{0.995^{k_0} \cdot n}_{=1}$$

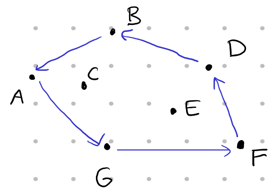
$$= k_0 + O(1) \leq O(\log n)$$

jede Iteration braucht $O(n)$
$\rightarrow$ Laufzeit $O(n \log n)$

Konvexe Hülle

konvexe Hülle $conv(S) = \bigcap_{\substack{S \subseteq C \subseteq \mathbb{R}^d \\ C \text{ konvex}}} C$ ← Schnitt aller konvexen Mengen, die S enthalten

Annahme: allgemeine Lage (keine drei Punkte auf einer Linie und keine zwei Punkte gleiche x-Koordinate)



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

→ wir können mit LinAlg in $O(1)$ bestimmen, ob ein Punkt p links von der Randkante qr liegt

Wie können wir jetzt daraus einen einfachen Algorithmus bauen? ($O(n^3)$)
→ teste alle möglichen $n(n-1)$ Paare auf Randkantendasein
→ für jeden Test jeweils alle verbleibenden $n-2$ Punkte überprüfen

JARVIS WRAP

q_0: Punkt mit kleinster x-Koordinate
↳ Ecke der konvexen Hülle

```
FINDNEXT(q)
Wähle p_0 ∈ P \ {q} beliebig
q_next ← p_0
for all p ∈ P \ {q, p_0} do
    if p rechts von q q_next then q_next ← p
return q_next
```

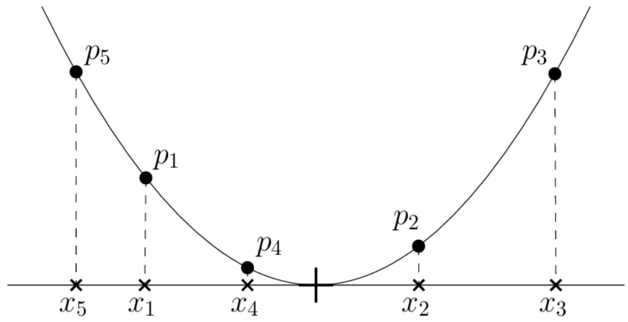
Warum müssen wir die for-Schleife nach q_{next}-Update nicht nochmal von vorne durchgehen?

```
JARVISWRAP(P)
h ← 0
p_now ← Punkt in P mit kleinster x-Koordinate
repeat
    q_h ← p_now
    p_now ← FINDNEXT(q_h)
    h ← h + 1
until p_now = q_0
return (q_0, q_1, ..., q_{h-1})
```

Laufzeit: $O(nh)$
↳ #Eckpunkte

Untere Schranke für ConvexHull-Problem

Kann man konvexe Hüllen in $t(n)$ berechnen, so kann man in $t(n) + O(n)$ Zeit sortieren.

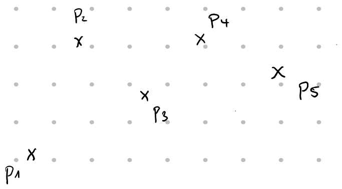


Lokal Verbessern

Annahme: allgemeine Lage und P aufsteigend sortiert nach x-Koordinate

Startpolygon: $(p_1, p_2, \dots, p_{n-1}, p_n, p_{n-1}, \dots, p_2)$

Idee Verbesserungsschritt: $\dots, p, p', p'', \dots$; falls p' links von pp'' liegt $\rightarrow$ entferne p' aus Folge




```
LocalRepair(p1, p2, ..., pn)
        ▷ setzt (p1, p2, ..., pn), n ≥ 3, nach x-Koordinate sortiert, voraus
q0 ← p1
h ← 0
for i ← 2 to n do                        ▷ untere konvexe Hülle, links nach rechts
    while h > 0 und qh links von q(h−1)pi do
        h ← h − 1
    h ← h + 1
    qh ← pi                   ▷ (q0, ..., qh) untere konvexe Hülle von {p1, ..., pi}
h' ← h
for i ← n − 1 downto 1 do                ▷ obere konvexe Hülle, rechts nach links
    while h > h' und qh links von q(h−1)pi do
        h ← h − 1
    h ← h + 1
    qh ← pi
return (q0, q1, ..., q(h−1)) ▷ Ecken der konvexen Hülle, gg. Uhrzeigersinn
```

Laufzeit: starten mit $2(n-1)$ Ecken und enden mit h Ecken
$\rightarrow O(n)$ Verbesserungen

$\rightarrow O(n \log n)$, da wir am Anfang sortieren müssen

QUIZFRAGEN

1) Betrachten Sie die Punktemenge $P = \{(-1,-1), (-1,1), (1,-1), (1,1)\}$

a) Ein Punkt $p = (p_1, p_2) \in \mathbb{R}^2$ erfüllt $out(p,P) = 1 \iff |p_1| \geq 1$ oder $|p_2| \geq 1$
→ Falsch

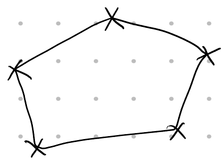
b) Wenn wir einen Punkt aus P herausnehmen, dann sind alle drei verbleibenden Punkte "essential".
→ Falsch

2) Für jede endliche Punktemenge $P \subset \mathbb{R}^2$, $|P| \geq 3$, gilt
$$\text{radius}(C(P)) \geq \max_{Q \subseteq P, |Q|=3} \text{radius}(C(Q))$$
→ Richtig, gilt mit Gleichheit

3) Sei $P \subseteq \mathbb{R}^2$ eine Menge von 50 Punkten in der Ebene. Wir wählen einen Punkt $x \in P$ zufällig und gleichverteilt. Wie gross ist die W'keit von $out(x, P \setminus \{x\}) = 1$ maximal?
→ $\frac{3}{50} = 0.06$

4) Seien $S \subseteq \mathbb{R}^2$ eine (möglicherweise unendliche) Untermenge von Punkten in der Ebene.

a) Wenn S endlich ist, dann ist conv(S) endlich
→ Falsch



b) Der Rand von conv(S) ist immer ein Polygon.
→ Falsch, Gegenbeispiel Kreis (s. unendlich viele Punkte)

5) Seien $S_1, S_2 \subseteq \mathbb{R}^2$, $C_1 = conv(S_1)$ und $C_2 = conv(S_2)$.
Dann $C_1 \cap C_2 = conv(S_1 \cap S_2)$.
→ Falsch