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 $\Pr[T \geq k]$:

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

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.