

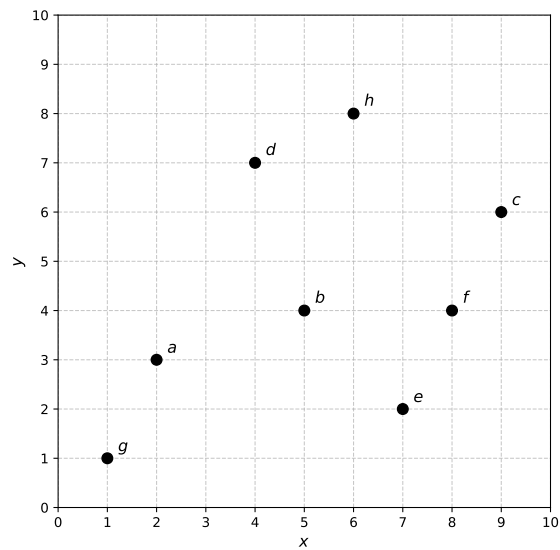
# Übungsblatt 11

Abgabe am 26.01.2025

## Aufgabe 1: KD-Trees (2+2+2 Punkte)

Gegeben sind die folgenden 8 Punkte in der Ebene:

Punkt	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
<i>x</i>	2	5	9	4	7	8	1	6
<i>y</i>	3	4	6	7	2	4	1	8



- a) Konstruieren Sie einen KD-Tree für diese Punktmenge. Beginnen Sie mit einer Aufteilung entlang der  $x$ -Achse und wechseln Sie in jeder Ebene die Achse. Teilen Sie jeweils am Median, sodass gleich viele Punkte links und rechts liegen (bei gerader Anzahl darf eine Seite einen Punkt mehr haben).

Zeichnen Sie:

- Die Trennlinien im Koordinatensystem
  - Den resultierenden Baum (mit Angabe der Trennachse und des Trennwerts in jedem inneren Knoten)
- b) Führen Sie eine Nearest-Neighbor-Suche für den Query-Punkt  $q = (2, 1)$  durch. Verwenden Sie dazu die euklidische Distanz  $d(p, q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$ .
- In welcher Reihenfolge werden die Knoten besucht?
  - Wie ändert sich der aktuelle "beste Kandidat" und dessen Distanz?
  - Welche Teilbäume werden nicht besucht (Pruning) und warum?
- c) Geben Sie eine Punktmenge und einen Query-Punkt an, sodass bei der Suche im entsprechenden KD-Tree *kein* Pruning möglich ist und alle Punkte besucht werden müssen.

## Aufgabe 2: Cloud Computing (2+2+2+2 Punkte)

Ein Startup betreibt eine App, bei der rechenintensive Anfragen bearbeitet werden müssen. Für jede Anfrage gibt es zwei Optionen:

- **Gratis-Server:** Die Berechnung dauert  $t$  Zeiteinheiten (der Wert  $t$  ist im Voraus unbekannt). Es entstehen keine Geldkosten.
- **Bezahl-Server:** Die Berechnung ist sofort fertig (1 Zeiteinheit) und kostet  $c = 5$  Euro.

Man kann die Berechnung zunächst auf dem Gratis-Server starten und jederzeit auf den Bezahl-Server wechseln. Die bereits auf dem Gratis-Server verstrichene Zeit ist dann jedoch verloren. Die *Gesamtkosten* einer Strategie sind definiert als:

$$\text{Kosten} = (\text{verbrauchte Zeit}) + (\text{Geldkosten in Euro})$$

Wir betrachten die Strategie  $s_k$ : *“Starte auf dem Gratis-Server. Falls die Berechnung nach  $k$  Zeiteinheiten noch nicht fertig ist, wechsle auf den Bezahl-Server.”*

- Sei  $\sigma_t$  eine Anfrage, deren Bearbeitung auf dem Gratis-Server genau  $t$  Zeiteinheiten dauert. Geben Sie die Kosten  $s_k(\sigma_t)$  der Strategie  $s_k$  an für die Fälle:
  - $t \leq k$  (die Berechnung wird auf dem Gratis-Server fertig)
  - $t > k$  (Wechsel auf den Bezahl-Server)
- Was sind die optimalen Offline-Kosten  $\text{OPT}(\sigma_t)$ ? Ab welchem Wert von  $t$  lohnt sich der Bezahl-Server für den Offline-Algorithmus?
- Bestimmen Sie den optimalen Wert  $k^*$ , der den kompetitiven Faktor der Strategie  $s_{k^*}$  minimiert.  
*Hinweis:* Berechnen Sie den kompetitiven Faktor für beide Fälle  $t \leq k$  und  $t > k$  separat und finden Sie das  $k^*$ , das den Worst-Case minimiert.
- Berechnen Sie den kompetitiven Faktor der optimalen Strategie  $s_{k^*}$ .

## Aufgabe 3: Treppensteigen (3 Punkte)

Sie wollen eine Treppe mit  $n$  Stufen hinaufsteigen. Dabei können Sie in jedem Schritt entweder eine, zwei oder drei Stufen gleichzeitig nehmen. Entwickeln Sie einen effizienten Algorithmus, der die Anzahl der verschiedenen Möglichkeiten berechnet die Treppe hinaufzusteigen.