

---

## Grundlagen: Algorithmen und Datenstrukturen

---

*Abgabetermin: In der jeweiligen Tutorübung*

### Hausaufgabe 1

Implementieren Sie den QuickSelect-Algorithmus.

Implementieren Sie in der Klasse `UISqsArray` in der Funktion `quickSelect` den QuickSelect-Algorithmus der das  $i$ -te Element in dem unsortierten Feld `A` liefert.

Verwenden Sie für Ihre Implementierung die auf der Übungswebseite bereitgestellten Klassen und verändern Sie für Ihre Implementierung *ausschließlich* die Klasse `UISqsArray`.

Achten Sie bei der Abgabe Ihrer Aufgabe darauf, dass Ihre Klasse `UISqsArray` heißt und auf den Rechnern der Rayhalle ([rayhalle.informatik.tu-muenchen.de](http://rayhalle.informatik.tu-muenchen.de)) mit der bereitgestellten Datei `main_qs` kompiliert werden kann. Anderenfalls kann eine Korrektur nicht garantiert werden. Achten Sie darauf, dass Ihr Quelltext ausreichend kommentiert ist.

Schicken Sie die Lösung per Email mit dem Betreff `[GAD] Gruppe <Gruppennummer>` an Ihren Tutor.

### Aufgabe 1

Beweisen Sie folgende in der Vorlesung benötigte Aussage: Ein fast vollständiger Binärbaum der Höhe  $h$  hat  $2^{h-1} \leq n \leq 2^h - 1$  Knoten (Die Wurzel hat Höhe 1).

*Definition;* Ein fast vollständiger Binärbaum der Höhe  $h$  ist ein vollständiger Binärbaum der Höhe  $h - 1$  dessen Knoten so angeordnet werden können, dass es auf dem Level  $h$  einen Knoten  $e$  mit dem Vaterknoten  $v$  gibt so dass gilt:

- Alle Knoten auf dem Level  $h - 1$  die „links“ von  $v$  stehen zwei Kinder haben
- und alle Knoten auf dem Level  $h - 1$  die „rechts“ von  $v$  stehen keine Kinder haben
- oder  $e$  ist die Wurzel.

### Aufgabe 2

Seien  $A$  und  $B$  nicht leere Mengen und  $\leq_A$  und  $\leq_B$  totale Ordnungen auf  $A$  bzw.  $B$ . Definieren Sie eine totale Ordnung auf  $A \times B$ .

Wozu brauchen wir den Begriff der totalen Ordnung im Zusammenhang der aktuellen Vorlesungsthemen Sortieren und Selektieren?

*Hinweis:* Eine lineare Ordnung  $\leq$  auf einer Menge  $X$  ist

- transitiv:  $\forall x, y, z \in X : x \leq y \wedge y \leq z \Rightarrow x \leq z$

- antisymmetrisch:  $\forall x, y \in X : x \leq y \wedge y \leq x \Rightarrow x = y$
- linear (total):  $\forall x, y \in X : x \leq y \vee y \leq x$
- reflexiv:  $\forall x \in X : x \leq x$  (folgt aus der Linearität)

### Aufgabe 3

Wir betrachten die Anzahl der Blocktransfers die beim externen Sortieren auftreten. Diese ist in der Vorlesung mit  $\mathcal{O}(\frac{n}{B} \log_{M/B} \frac{n}{M})$  hergeleitet worden.

- Zeigen Sie: Der Algorithmus für das externe Sortieren muss unter normalen Umständen (realistische Werte für  $M$  und  $B$ ) für noch nicht einmal jedes Element eine Lese- oder Schreiboperation ausführen.

Gehen Sie dabei wie folgt vor: Gehen Sie von der asymptotischen Anzahl der Blocktransfers aus und geben Sie eine Bedingung für  $n$  an, so dass mindestens  $n$  Lese- oder Schreiboperation auf dem Hauptspeicher ausgeführt werden müssen. Argumentieren Sie dann warum diese Bedingung meist nicht zu erfüllen ist, oder wie  $M$  und  $B$  zu wählen ist, damit diese Bedingung realistisch erfüllbar ist.

- Man kann für externes Sortieren eine untere Schranke von  $\Omega(\frac{n}{B} \log_{M/B} \frac{n}{B})$  benötigten Blocktransfers zeigen. Ist damit der in der Vorlesung gezeigte Algorithmus asymptotisch optimal?