



5. Übungsblatt zu Grundzüge von Algorithmen und Datenstrukturen, WS 15/16

Prof. Markus Bläser

<http://www-cc.cs.uni-saarland.de/course/50/>

Abgabe: 26. November 2015, 12:00 Uhr

Aufgabe 5.1 In der Vorlesung wurde der Algorithmus von Blum, Floyd, Pratt, Rivest und Tarjan vorgestellt, der das kt -größte von n Elementen in $O(n)$ Schritten findet. Untersuchen Sie, ob die folgenden Varianten ebenfalls lineare Laufzeit haben.

- Anstatt in 5er-Gruppen wird in 3er-Gruppen aufgeteilt.
- Anstatt in 5er-Gruppen wird in 3er-Gruppen aufgeteilt. Die Mediane dieser 3er-Gruppen werden wieder in 3er-Gruppen aufgeteilt, von denen dann die Mediane bestimmt werden. Mit der Menge dieser Mediane wird rekursiv fortgefahren.

Aufgabe 5.2 a) Arbeiten Sie das Kapitel im Skript durch, in dem Stacks und Queues erläutert werden.

- Wir verallgemeinern die Pop-Operation bei einem Stack: $\text{Mulpop}(S, k)$ entfernt k Elemente auf einmal vom einem Stack S . Sind weniger als k Elemente vorhanden, so werden alle entfernt. Die Kosten einer Mulpop-Anweisung entspricht der Anzahl der tatsächlich entfernten Elemente. Die Kosten einer Push-Operation sei 1. Zeigen Sie: Führt man eine beliebige Folge von n Push- oder Mulpop-Anweisungen auf einem Stack aus, der anfangs mit s Elementen gefüllt ist, so sind die Gesamtkosten hierfür $\leq 2n + s$. (Dies bedeutet, dass obwohl eine einzelne Mulpop-Anweisung beliebig teuer sein kann, die durchschnittlichen Kosten pro Anweisung $\leq 2 + s/n$ sind.)

Aufgabe 5.3 Gegeben sei eine Menge U von n Elementen. Wir wollen eine Menge von disjunkten Teilmengen $S_1, \dots, S_\ell \subseteq U$ verwalten. Jede Menge S_i wird durch ein ausgezeichnetes Element in der Menge repräsentiert, dem sogenannten Repräsentanten. Da die Mengen disjunkt sind, sind Repräsentanten paarweise verschieden. Die Datenstruktur soll folgende Operationen unterstützen:

Make-set(x): Erschaffe eine neue Menge bestehend aus dem Element x . x darf dabei in keiner anderen Menge bereits vorkommen. Der Repräsentant dieser Menge ist x .

Union(x, y): Berechnet die Vereinigung der Mengen mit den Repräsentanten x und y . Dabei sind x und y verschieden. Die alten Mengen werden zerstört.

Find(x): Gibt den Repräsentanten der Menge zurück, zu der x gehört. Gehört x zu keiner Menge, so wird NULL zurück gegeben.

Implementieren Sie obige Datenstruktur mit Hilfe von verketteten Listen und analysieren Sie die Laufzeit.