

3. Übung zur Informatik I

Abgabe am Freitag, den 02.05, 8:00 Uhr im Fach im Keller des Mathematischen Instituts (Weyertal 86-90)

Bitte schreiben Sie auf Ihre Abgabe Ihren Namen und Ihre Übungszeit. Es werden nur handschriftliche Abgaben akzeptiert.

Aufgabe 1:

3 Punkte

In einem Kopfbahnhof mit drei Gleisen befinden sich in den Gleisen S_1 und S_2 Eisenbahnwaggons, die für die Zielbahnhöfe Köln und Hamburg bestimmt sind, aber leider durcheinander stehen. Das Gleis S_3 sei leer. Betrachten Sie S_1, S_2, S_3 als Stacks, und schreiben Sie ein Rangierprogramm unter der Verwendung der unten angegebenen Prozeduren, das die Wagen so umordnet, daß anschließend in S_1 alle Waggons nach Köln und in S_2 alle Waggons nach Hamburg stehen.

- $push(S : Stapel; X : Waggon)$
{stellt Waggon X auf S ab}
- $pop(S : Stapel) : Waggon$
{liefert den vordersten Waggon von S und entfernt ihn von S , falls S nicht leer ist; sonst Fehler}
- $top(S : Stapel) : Zielbahnhof$
{liefert den Zielbahnhof des ersten Waggons in S , ohne den Waggon zu entfernen}
- $leer(S : Stapel) : Boolean$
{ $true$, falls S leer ist, sonst $false$ }

Aufgabe 2:

4 + 2 + 4 Punkte

Um das Maximum und das Minimum einer n -elementigen Menge S zu bestimmen, kann man das folgende Divide-and-Conquer Verfahren anwenden:

1. Divide: Zerlege die Menge S in zwei Teilmengen S_1 und S_2 mit $|S_1| = \lceil |S|/2 \rceil$ und $|S_2| = \lfloor |S|/2 \rfloor$.
2. Conquer: Wende das Verfahren rekursiv an, um $\max(S_1), \min(S_1), \max(S_2)$ und $\min(S_2)$ zu berechnen.
3. Berechne daraus $\max(S)$ und $\min(S)$.

- (a) Formulieren Sie zu diesem Algorithmus ein rekursives Programm in der Pseudo-Programmiersprache.
- (b) $V(n)$ sei die Anzahl der Vergleiche, die der obenstehende Algorithmus ausführt, um das Maximum und das Minimum einer n -elementigen Menge zu erhalten. Zeigen Sie, daß für $n = 2^k$ gilt:

$$V(n) = \frac{3}{2}n - 2$$

- (c) Mit Hilfe von Stacks kann man rekursive Probleme iterativ abarbeiten, indem man Teilprobleme, die zunächst zurückgestellt werden, auf den Stack stapelt und später weiter bearbeitet, bis der Stapel leer ist. Schreiben Sie ein iteratives Programm in der Pseudo-Programmiersprache, das das obige Verfahren realisiert. Benutzen Sie dazu einen Stack und die Stackoperationen *push* und *pop*, ohne sich Gedanken über die Repräsentation der Menge S durch konkrete Datenstrukturen zu machen. Die Mengenoperationen $Vereinige(S_1, S_2, S_1 \cup S_2)$ und $Partitioniere(S, S_1, S_2)$ dürfen verwendet werden.

Aufgabe 3:

1 + 6 Punkte

- (a) Bei der Durchmusterung eines Binärbaumes in Präordnung werden die Knoten in der Reihenfolge

O B T H K E N S U C

durchlaufen, während sich bei der symmetrischen Durchmusterung des Baumes die Reihenfolge

H T B E K N O U S C

ergibt. Zeichnen Sie einen Binärbaum, dessen Durchmusterung des Baumes die Zeichenfolgen liefert. Ist die Lösung eindeutig?

- (b) Zeigen Sie, daß man einen binären Baum immer rekonstruieren kann, wenn die Durchmusterungsreihenfolgen in Präordnung und der symmetrischen Durchmusterung bekannt sind.
Was passiert, wenn man Präordnungs- und Postordnungsreihenfolge beziehungsweise Postordnungsreihenfolge und symmetrische Durchmusterungsfolge kennt?