

Probeklausur zum zweiten Teil der Vorlesung
Informatik I

Prof. Dr. J. Rethmann
SS 2006

Aufgabe 1: (Quickies)

(6 Punkte)

Welche der folgenden Aussagen sind richtig bzw. falsch?

Achtung: Für jede falsche Antwort wird eine richtige Antwort nicht bewertet. Für eine unbeantwortete Frage werden keine Punkte abgezogen.

- | | richtig | falsch |
|--|--------------------------|--------------------------|
| (a) die average-case Laufzeit von Quick-Sort ist in $O(n^2)$ | <input type="checkbox"/> | <input type="checkbox"/> |
| (b) die worst-case Laufzeit von Merge-Sort ist in $\Omega(n)$ | <input type="checkbox"/> | <input type="checkbox"/> |
| (c) $T(n) = 8T(n/2) + \Theta(n^2) \in \Omega(n^2)$, wobei $T(1) \in O(1)$ | <input type="checkbox"/> | <input type="checkbox"/> |
| (d) $O(2^{n+m}) = O(2^{n^m})$ | <input type="checkbox"/> | <input type="checkbox"/> |
| (e) $O(\log_2(n)) = O(\log_3(n))$ | <input type="checkbox"/> | <input type="checkbox"/> |
| (f) $f_1 \in O(g_1)$ und $f_2 \in O(g_2) \Rightarrow f_1 + f_2 \in O(g_1 + g_2)$ | <input type="checkbox"/> | <input type="checkbox"/> |

Aufgabe 2: (Merge Sort)

(8 Punkte)

Sortieren Sie die Zahlenfolge 11, 4, 2, 6, 9, 1, 2, 7, 0, 9, 12, 3, 6, 14, 16, 5 mittels Merge Sort. Geben Sie die resultierende Zahlenfolge nach jedem Merge-Schritt an.

Aufgabe 3: (Exponentielle Suche)

(8 Punkte)

Suchen Sie den Wert 93 mittels exponentieller Suche in der Folge:

1, 3, 5, 7, 9, 11, 25, 37, 49, 61, 73, 85, 87, 91, 93, 95, 99, ...

Bestimmen Sie zunächst die linke und rechte Grenze des zu durchsuchenden Bereichs. Durchsuchen Sie anschließend diesen Bereich mittels binärer Suche, wobei in jedem Schritt die linke und rechte Grenze sowie der Index des zu prüfenden Elements anzugeben sind.

Aufgabe 4: (Hash-Verfahren)

(6 Punkte)

Fügen Sie die Werte 27,2,14,6,4,16,24,5 in eine Hash-Tabelle der Größe $m = 11$ ein. Die Hash-Funktion sei $h(k) = k \bmod m$.

Auftretende Kollisionen sollen durch lineares Sondieren aufgelöst werden. Geben Sie die Hash-Tabelle nach jedem Einfügen eines Wertes an.

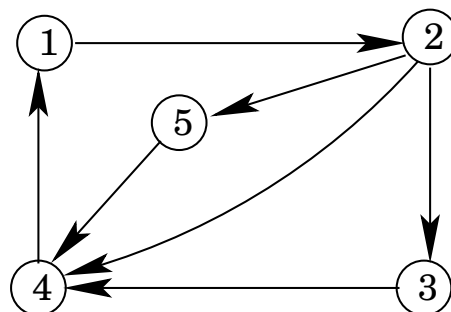
| | | | | | | | | | | |
|----|--|--|--|--|--|--|--|--|--|--|
| 27 | | | | | | | | | | |
| 2 | | | | | | | | | | |
| 14 | | | | | | | | | | |
| 6 | | | | | | | | | | |
| 4 | | | | | | | | | | |
| 16 | | | | | | | | | | |
| 24 | | | | | | | | | | |
| 5 | | | | | | | | | | |

Aufgabe 5: (Tiefensuche)

(12 Punkte)

Gegeben sei der unten stehende Graph. Starten Sie eine Tiefensuche bei Knoten 1. Geben Sie die dfb- (depth first begin) und die dfe-Nummerierung (depth first end) an, und kennzeichnen Sie jede Kante als Baum-, Vorwärts-, Rückwärts- bzw. Querkante.

Die von einem unbesuchten Knoten ausgehenden Kanten sollen in aufsteigender Reihenfolge der Nummern ihrer Endknoten rekursiv besucht werden.



Aufgabe 6: (Kürzeste Wege)

(12 Punkte)

Bestimmen Sie mit Hilfe von Dijkstras-Algorithmus alle kürzesten Wege im unten stehenden Graphen vom Startknoten 1 aus.

- Geben Sie nach jedem Schritt (d.h. dem Korrigieren der Distanzwerte aller Nachbarknoten des aktuellen Knotens) die Datenstruktur (Priority Queue) Q an.
- Schreiben Sie die aktuellen Distanzwerte jeweils an die Knoten.

