



Übungsblatt 12

Aufgabe T42

Beim Divide-and-Conquer-Algorithmus für das Finden der nächsten Nachbarn ist es ja ziemlich unangenehm, den mittleren Streifen noch einmal extra zu betrachten. Ein Schlaumeier schlägt folgenden Algorithmus vor:

Falls es weniger als, sagen wir, zehn Punkte gibt, dann lösen wir das Problem in konstanter Zeit indem wir alle Paare betrachten. Andernfalls teilen wir die Punkte horizontal in etwa drei gleich große Mengen A , B und C auf, anstatt in nur zwei. Dann rufen wir den Algorithmus zweimal rekursiv auf: Einmal auf $A \cup B$ und einmal auf $B \cup C$. Von beiden Antworten ist die kleinere dann die korrekte Lösung.

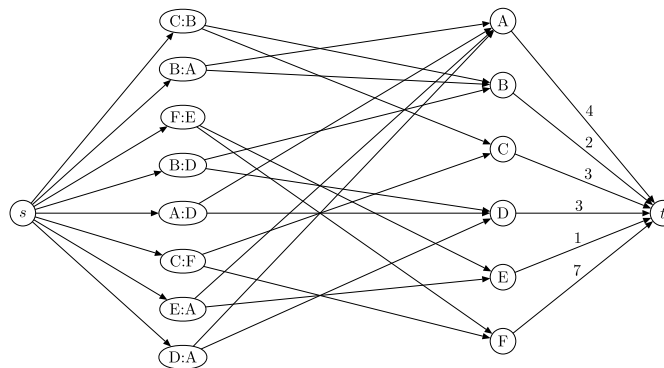
Ist der Algorithmus korrekt? Falls nein, kann der Fehler leicht behoben werden?

Aufgabe T43

Wir sind mitten in einem Fußballturnier. Außer unserer Mannschaft G gibt es noch die Mannschaften A, \dots, F . Bei einem Gewinn bekommt die siegreiche Mannschaft zwei Punkte, bei einem Unentschieden beide einen Punkt. Wir möchten wissen, ob unsere Mannschaft das Turnier noch gewinnen kann. Unter der Annahme, dass wir alle unserer vier eigenen Spiele gewinnen, geht das nur wenn A höchstens noch vier Punkte gewinnt, B noch höchstens zwei, und so weiter.

Die anderen Mannschaften spielen insgesamt noch acht Spiele. Bei einem Gewinn bekommt die siegreiche Mannschaft zwei Punkte, bei einem Unentschieden beide einen Punkt. Wie kann das abgebildete Flussnetzwerk uns sagen, ob wir noch eine Chance haben? Die unbeschrifteten Kanten haben Kapazität 2.

Team	Punkte
E	16
B	15
C	14
D	14
A	13
F	10
G	10



Aufgabe T44

Ermitteln Sie einen maximalen Fluss für das Netzwerk auf der folgenden Seite, wobei die Quellen links und die Senken rechts sind. Beweisen Sie die Maximalität durch Angabe eines gleichgroßen Schnitts. Alle Kapazitäten sind 1.

Aufgabe H34 (10 Punkte)

Geben Sie eine Folge von Union- und Find-Operationen an, die zu einem Baum der Höhe vier führt (vier Knoten übereinander). Verwenden Sie dabei sowohl die Rangheuristik als auch Pfadkompression.

Nehmen Sie an, dass bei der Operation $union(A, B)$ der Baum B in A eingehängt wird, falls beide gleichen Rang haben. Geben Sie das Zwischenergebnis nach jeder Operation an.

Aufgabe H35 (12 Punkte)

Entwerfen Sie einen Algorithmus, der in einem gegebenen Wort mit Länge n das längste Unterwort findet, das ein Palindrom ist. Die Laufzeit des Algorithmus sollte höchstens $O(n^2)$ sein. Sie dürfen einen deterministischen oder einen randomisierten Algorithmus erfinden.

Aufgabe H36 (8 Punkte)

Liste Sortierte Liste Array Sortiertes Array Binärer Suchbaum AVL-Baum Splay-Tree Treap Skip-List Hashtabelle Min-Heap

Randomisiert											
Einfügen											
Suchen											
Löschen											
Minimum											
Maximum											
Sort. Ausgeben											

Beantworten Sie die Fragen für alle Datenstrukturen bzw. geben Sie eine obere Schranke für den Aufwand an. Gehen Sie davon aus, dass die Anzahl der enthaltenden Elemente n beträgt. Für Laufzeiten tragen Sie eine Funktion $f(n)$ in die Tabelle ein, um eine Laufzeit von $O(f(n))$ auszudrücken. Bei randomisierten Datenstrukturen ist die erwartete obere Schranke gemeint, bei amortisierten Analysen die amortisierte Laufzeit.

