Abteilung Algorithmik
Institut für Betriebssysteme und Rechnerverbund **WS 11/12**
TU Braunschweig

Prof. Dr. Sándor Fekete
Dr. Christiane Schmidt

Algorithmen und Datenstrukturen
Übung 5 vom 25.01.2012

Abgabe der Lösungen am Mittwoch, den 08.02.12, vor der Abteilung *Algorithmik*.
Bitte die Blätter vorne deutlich mit eigenem **Namen** und **Gruppennummer** versehen!

Aufgabe 1 (Klausurvorbereitung):
Gib Deinen Namen (Format: Nachname, Vorname), Matrikelnummer und Studiengang (mit Zusatz Bachelor, Master, Diplom!) *leserlich* an.
Diese Angaben brauchen wir für die Weiterleitung der Klausurergebnisse, also gebt euch Mühe `;-)`.

(2 Punkte)

Aufgabe 2 (AVL-Bäume):

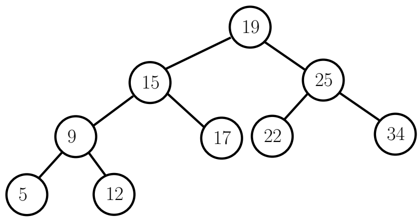

Abbildung 1: Der AVL-Baum T.

Gegeben sei der AVL-Baum T aus Abbildung 1. Füge nacheinander die Elemente 14 und 20 ein, so dass T ein AVL-Baum bleibt; gib den Baum nach jeder Einfüge-Operation an. (Hinweis: Nach *jedem* Einfügen soll der geänderte Baum ein AVL-Baum sein. Zum Schluss sollen beide Zahlen eingefügt sein.)

(15 Punkte)

Aufgabe 3 (Konvexe Hüllen):
Die *konvexe Hülle* einer Menge Q von Punkten, $CH(Q)$, ist das kleinste konvexe Polygon P, für das sich jeder Punkt aus Q entweder auf dem Rand von P oder in seinem Inneren befindet. Betrachte Abbildung 2 für ein Beispiel.
Ein möglicher Algorithmus zur Berechnung der konvexen Hülle verwaltet einen Stack S aus Punktkandidaten. Er legt jeden Punkt der Eingabemenge Q (wir nehmen an $|Q| \geq 3$) zunächst auf den Stapel und entfernt jeden Punkt, der keine Ecke der konvexen Hülle $CH(Q)$ ist, letztendlich wieder vom Stapel. Wenn der Algorithmus terminiert, enthält S genau die Ecken von $CH(Q)$ entgegen dem Uhrzeigersinn ihres Auftretens auf dem Rand. Der Algorithmus ruft die Funktionen TOP(S), die den obersten Punkt im Stack S zurückgibt, ohne den Stapel zu verändern, und NEXT-TO-TOP(S) auf, die den darunter liegenden Punkt im Stack S ohne Veränderung von S zurückgibt.

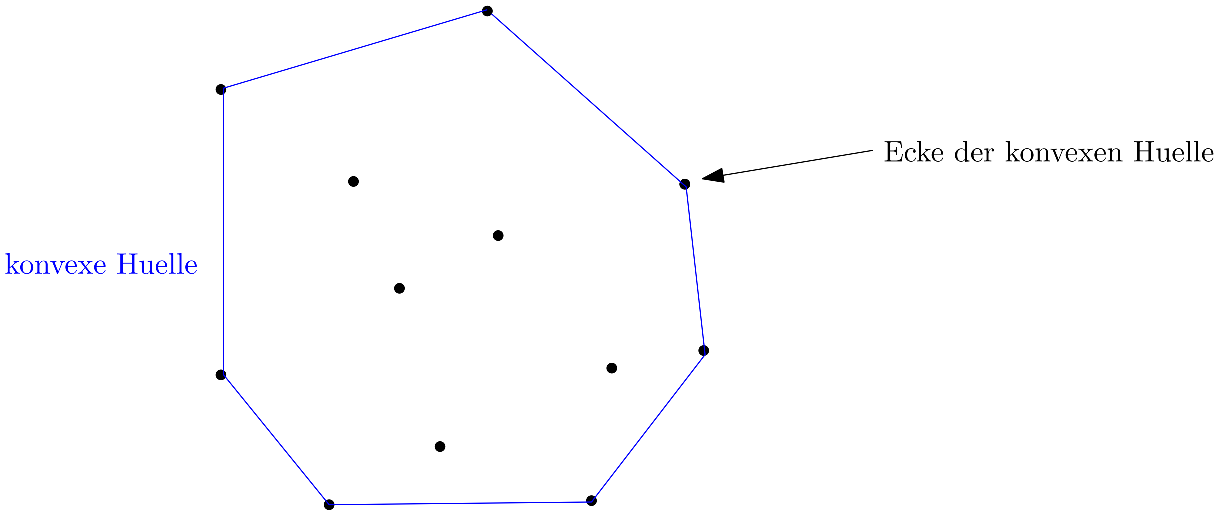

Abbildung 2: Beispiel für eine konvexe Hülle.

Zunächst wird der Punkt p_0 von Q mit minimaler y-Koordinate bestimmt (gibt es mehrere solche, der am weitesten links liegende solche Punkt). Die anderen Punkte $(p_1, \ldots, p_n)$ werden nach dem Polarwinkel relativ zu p_0 gegen den Uhrzeigersinn geordnet (die Horizontale als Nullinie nutzend).
Zu Beginn werden die ersten drei Punkte auf den Stack gelegt. Solange der von NEXT-TO-TOP(S), TOP(S) und p_i (dem aktuell betrachteten Knoten, startend mit dem vierten) gebildete Winkel < 180 ist (siehe Abbildung 3), wird der oberste Knoten jeweils vom Stack entfernt. Bevor der nächste Knoten betrachtet wird (nächste nach der oben gegebenen Ordnung), wird p_i oben auf den Stapel gelegt.

a) Formuliere den oben beschriebenen Algorithmus in Pseudocode.

b) Wende den Algorithmus auf die Punktmenge Q aus Abbildung 4 an.

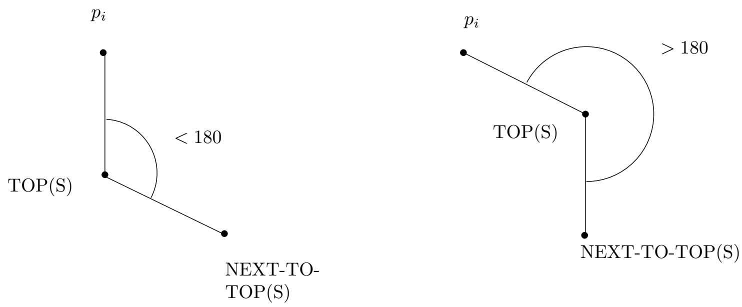

Abbildung 3: Der von NEXT-TO-TOP(S), TOP(S) und p_i gebildete Winkel.

(10+3 Punkte)

Aufgabe 4 (Mergesort):

Sortiere die Sequenz $(12, 8, 25, 14, 9, 11, 11, 4)$ mit Hilfe von Mergesort. Gib die Zwischenschritte in geeigneter Form an.

(15 Punkte)

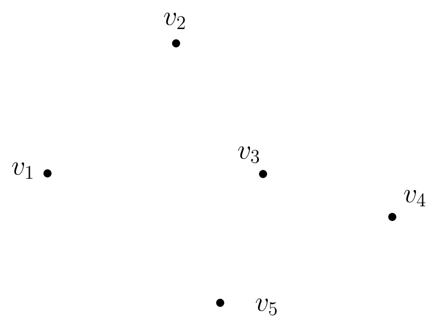

Abbildung 4: Die Punktmenge Q.

Aufgabe 5 (Mastertheorem):

a) Bestimme mit Hilfe des Mastertheorems das asymptotische Wachstum der Rekursion $U(n) = U(\frac{n}{2}) + 8 \cdot U(\frac{n}{3}) + 3n^3$.
Bestimme die Werte aller im Mastertheorem auftretenden Parameter.

b) Bestimme mit Hilfe des Mastertheorems das asymptotische Wachstum der Rekursion $V(n) = 64 \cdot V(\frac{n}{4}) + 17n^2$.
Bestimme die Werte aller im Mastertheorem auftretenden Parameter.

c) Bestimme mit Hilfe des Mastertheorems das asymptotische Wachstum der Rekursion $T(n) = 10 \cdot T(\frac{n}{5}) + 2 \cdot T(\frac{n}{2}) + 10 \cdot T(\frac{n}{10}) + 25n^2$.
Bestimme die Werte aller im Mastertheorem auftretenden Parameter.

(5+5+5 Punkte)