Aufgaben Schmid

Aufgabe 1: (Die Ackermannfunktion – 12 P)
Die Ackermannfunktion wird rekursiv definiert durch

$$a(0,m) = m+1 \qquad a(n,0) = a(n-1,1) \qquad a(n,m) = a(n-1, a(n,m-1))$$

Als Beispiel berechnen wir
a(1,0) = a(0,1) = 1+1 = 2
a(1,1) = a(0,a(1,0)) = a(0,2) = 3

Berechnen Sie a(1,2) (2 P).
Implementieren Sie eine Funktion int a, welche die Ackermannfunktion berechnet (10 P).

Aufgabe 2: (Heapsort – 11 P)
Ein Heap kann als Array oder als Baum dargestellt werden. Stellen Sie folgenden Heap (abhängig von x) als Baum dar (1, 10, 4, 11, 20, x, 6, 30, 40, 50, 60, 7, 13, 8) (3 P).
Welche Werte kann x annehmen, wenn bekannt ist, dass x vom Typ Integer ist und alle Zahlen im Heap verschieden sind (3 P)?
Führe einen Heapsortschritt aus (DeleteMin). Wie sieht der reorganisierte Heap aus (3 P)?
Gibt es Suchbäume, die gleichzeitig Minheaps sein können (2 P)?

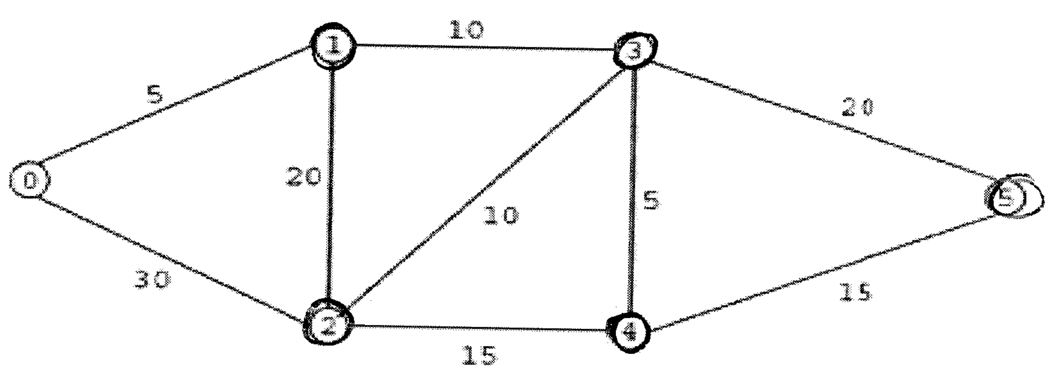
Aufgabe 3: (Suchbäume / Sortieren – 9 P)
Die Folge (3,1,8,2,6,9,4,7,5) soll mit Hilfe von Baumsortieren sortiert werden. Fügen Sie die Sequenz zunächst sukzessive in einen Suchbaum ein und geben Sie den Ergebnissuchbaum an (3 P).
Wie können Sie mit Hilfe des Suchbaumes die Folge sortieren (1 P)?
Löschen Sie jetzt das Element 3 im Suchbaum und reorganisieren Sie den Suchbaum. Zum Ersetzen wählen Sie den Inordernachfolger (3 P).

Kann es passieren, dass nach dem ersten Bubblesortschritt das zweitgrößte Element an erster Position steht. Wenn ja, wie und wenn nein, warum nicht (2 P)?

Aufgabe 4: (Suchen – 6 P)
Begründen Sie kurz, warum bei binärer Suche der Aufwand im Mittel nahe beim höchsten Aufwand liegt (3 P). Gegeben sei folgendes Feld int a[10] mit dem Inhalt (2,3,4,5,6,7,8,9,13,20). An welcher Position würde man das Element 6 mit Interpolationssuche zuerst suchen und welches Element würde die Suche stattdessen vorfinden (3 P)?

Aufgabe 5: (Baumdurchläufe – 3 P)
Welche Eigenschaft muss ein Binärbaum haben, damit sein Inorderdurchlauf absteigend sortiert ist (3 P)?

Aufgabe 6: (Graphdurchläufe – 6 P)
Gegeben sei folgender gewichteter ungerichteter Graph. ‚Ungerichtet' bedeutet, dass mit der Kante (0,1) mit dem Gewicht 5 automatisch die Kante (1,0) wieder mit dem Gewicht 5 im Graph existiert. Führen Sie eine Tiefensuche und eine Breitensuche beginnend am Knoten 0 durch. Die Reihenfolge der Knoten beim ‚Pushen' (bzw. ‚Queuen') sei dabei von oben nach unten – das heißt zuerst wird der Knoten mit der höchsten Nummer gepusht, dann der mit der zweithöchsten etc.

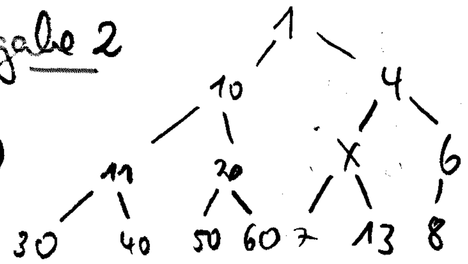


Aufgabe 1 $A(0,3) = 3+1 = 4$ (2)

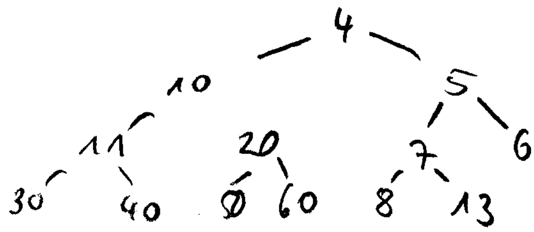
```
int A (int n, int m) {
 if (n==0){ return (m+1);}
 if (m==0){ return (A(n-1,1));}
 return (A(n-1), A(n,m-1))); }
```
(10)

Aufgabe 2

(3)



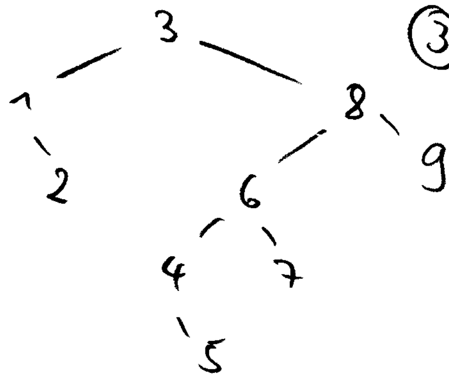
$4 < x < 7$ und $x \in \mathbb{Z}$ und $x \neq 6 \Rightarrow x = 5$ (3)



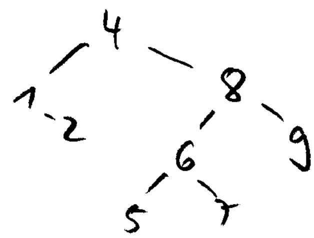
(3)

Der Suchbaum muß einelementig sein zB 1 sonst geht es nicht (2)

Aufgabe 3



(3) Ausgabe inorder ist sortiert (1)



(3)

Ja, wenn Max ganz links steht (2)

Aufgabe 4 Wenn das Feld $2^k - 1$ Elemente hat, so werden 2^{k-1} Elemente (mehr als die Hälfte) nach k Schritten gefunden. (3)

$P = \frac{(6-2)(10-1)}{20-2} + 1 = 3$ (von 1 ab gezählt) $A[3] = 4$ (3)

Aufgabe 5 x, L, R — im Unterbaum L stehen nur Elemente größer (oder $\geq$) x in R kleiner (oder $\leq$) (3)

Tiefensuche 0 1 2 3 4 5

Breitensuche 0 2 1 4 3 5

(6)