Technische Universität Wien
Institut für Computergraphik und Algorithmen
Arbeitsbereich für Algorithmen und Datenstrukturen

186.172 Algorithmen und Datenstrukturen 1 VL 4.0

Nachtragstest SS 2010

30. Juni 2010

Machen Sie die folgenden Angaben bitte in deutlicher Blockschrift:

Nachname: ______ Vorname: ______

Matrikelnummer: ______ Studienkennzahl: ______

Anzahl abgegebener Zusatzblätter: ______

Legen Sie bitte Ihren Studentenausweis vor sich auf das Pult.

Sie können die Lösungen entweder direkt auf die Angabeblätter oder auf Zusatzblätter schreiben, die Sie auf Wunsch von der Aufsicht erhalten. Es ist nicht zulässig, eventuell mitgebrachtes eigenes Papier zu verwenden.

Die Verwendung von Taschenrechnern, Mobiltelefonen, Skripten, Büchern, Mitschriften, Ausarbeitungen oder vergleichbaren Hilfsmitteln ist unzulässig.

Die Arbeitszeit beträgt 55 Minuten.

	A1:	A2:	A3:	Summe:
Erreichbare Punkte:	20	14	16	50
Erreichte Punkte:				

Viel Erfolg!

Aufgabe 1.A: ADTS und Suchen **(20 Punkte)**

a) (12 Punkte) Schreiben Sie detaillierten Pseudocode für die Funktion $merge(L_1, L_2)$. Dabei sind die Parameter L_1 und L_2 jeweils Zeiger auf das erste Element einer aufsteigend sortierten, azyklischen, einfach verketteten Liste. Als Rückgabe soll die Funktion $merge(L_1, L_2)$ einen Zeiger auf das erste Element einer aufsteigend sortierten, einfach verketteten Liste liefern, die alle Elemente aus der Liste L_1 und L_2 enthält.

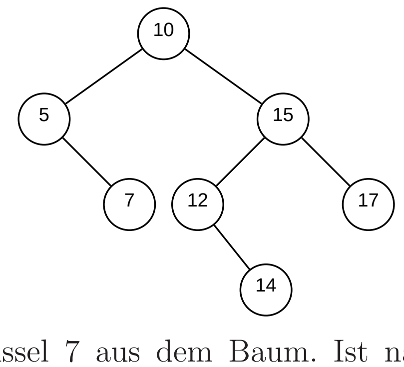
b) (8 Punkte)

- Zeichnen Sie einen AVL Baum, bei dem eine Postorder-Durchmusterung folgendes Resultat liefert:

$$< 33, 40, 55, 65, 70, 60, 50 >$$

  Ist der AVL-Baum mit der oben angeführten Postorder-Durchmusterung eindeutig rekonstruierbar? Begründen Sie Ihre Antwort.

- Gegeben sei weiters folgender AVL-Baum:

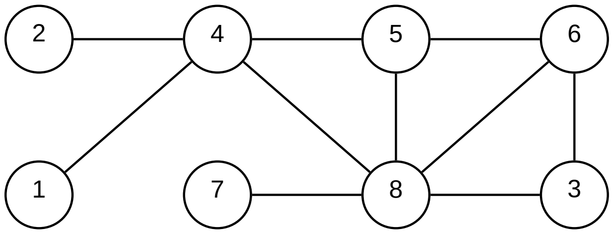


  Löschen Sie den Schlüssel 7 aus dem Baum. Ist nach dem Entfernen des Schlüssels 7 die AVL-Baum Eigenschaft verletzt? Falls ja, führen Sie alle notwendigen Reorganisationsmaßnahmen durch, um erneut einen gültigen AVL-Baum zu erhalten, und zeichnen Sie den resultierenden AVL-Baum.

Aufgabe 2.A: Graphen und Optimierung (14 Punkte)

a) (11 Punkte)

Gegeben ist ein ungerichteter, zusammenhängender Graph $G = (V, E)$:



Gegeben ist ferner folgender Algorithmus:

Algorithmus Was-bin-ich?

Input: Graph $G = (V, E)$
Output: Kantenmenge F und Feld $X[1 \dots |V|]$
$F = \{\};$
$X[1 \dots |V|] = (-1, -1, \dots, -1);$
A(4);

Prozedur A(v)

$X[v] = X[v] + 1;$
for all Knoten $w \in N(v)$ in der gegebenen Ordnung **do**
  if $X[w] < 0$ **then**
    $F = F \cup (v, w);$
    A(w);
  end if
  $X[w] = X[w] + 1;$
end for

- Welche Kanten liefert der Algorithmus für den oben angeführten Graphen in F zurück? Markieren Sie diese Kanten im Graphen. Geben Sie weiters das Feld X nach der Ausführung des Algorithmus an.
- Beschreiben Sie in einem Satz, was dieser Algorithmus allgemein für einen beliebigen ungerichteten, zusammenhängenden Graphen macht und in F bzw. X zurückliefert.
- Welche Laufzeit ergibt sich bei diesem Algorithmus im Worst-Case, in Θ-Notation, in Abhängigkeit der Anzahl der Knoten $|V|$ und Kanten $|E|$, für einen beliebigen ungerichteten, zusammenhängenden Graphen, wenn dieser in Form von Adjazenzlisten gespeichert ist?

b) (3 Punkte)

Welche aus der Vorlesung bekannten Datenstrukturen und Verbesserungen werden benötigt, damit der Gesamtaufwand des Algorithmus von Kruskal $O(|E| \log |E|)$ nicht übersteigt? Begründen Sie Ihre Antwort!

Aufgabe 3.A: Sortierverfahren (16 Punkte)

a) (8 Punkte) Welche in der Vorlesung behandelten Sortierverfahren eignen sich am Besten für die unten angeführten Folgen. Begründen Sie Ihre Antwort.

- Eine beliebige Folge soll absteigend sortiert werden.
- In eine bereits absteigend sortierte Folge wird ein beliebig großes Element am Ende angehängt. Die neu entstandene Folge soll nun erneut absteigend sortiert werden.
- Eine sehr große Anzahl an Rechnungen, die jeweils mit einer Jahreszahl und einer 5-stelligen Rechnungsnummer versehen sind, soll sortiert werden.
- Bei einem Transport sollen schwere Pakete aufsteigend nach ihrer Bezeichnung sortiert werden. Schlüsselbewegungen sind dadurch wesentlich aufwendiger als Schlüsselvergleiche. Die Länge der Bezeichnung eines Paketes ist nicht bekannt.

b) (8 Punkte) Beweisen Sie, dass für die im Folgenden definierte Funktion $f(n)$ die Beziehung $f(n) = O(n^2)$ gilt.

Verwenden Sie für Ihren Beweis die Konstante $c = 2$ und wählen Sie einen geeigneten Wert für n_0. Beachten Sie, dass für einen gültigen Beweis auch die formale Definition angegeben werden muss.

$$f(n) = \begin{cases} n^2 + 100 \cdot n, & \text{falls } n > 10 \\ 10 \cdot n^2 + n + 5, & \text{sonst} \end{cases}$$

Kreuzen Sie anschließend in der folgenden Tabelle die zutreffenden Felder für die oben angeführte Funktion $f(n)$ an:

$f(n)$ ist	$\Theta(.)$	$O(.)$	$\Omega(.)$	keines
$n\sqrt{n}$				
n^3				
n^2				
$n \log_2(2^n)$				

Technische Universität Wien
Institut für Computergraphik und Algorithmen
Arbeitsbereich für Algorithmen und Datenstrukturen

186.172 Algorithmen und Datenstrukturen 1 VL 4.0

Nachtragstest SS 2010

30. Juni 2010

Machen Sie die folgenden Angaben bitte in deutlicher Blockschrift:

Nachname: ________ Vorname: ________

Matrikelnummer: ________ Studienkennzahl: ________

Anzahl abgegebener Zusatzblätter: ____

Legen Sie bitte Ihren Studentenausweis vor sich auf das Pult.

Sie können die Lösungen entweder direkt auf die Angabeblätter oder auf Zusatzblätter schreiben, die Sie auf Wunsch von der Aufsicht erhalten. Es ist nicht zulässig, eventuell mitgebrachtes eigenes Papier zu verwenden.

Die Verwendung von Taschenrechnern, Mobiltelefonen, Skripten, Büchern, Mitschriften, Ausarbeitungen oder vergleichbaren Hilfsmitteln ist unzulässig.

Die Arbeitszeit beträgt 55 Minuten.

	B1:	B2:	B3:	Summe:
Erreichbare Punkte:	16	14	20	50
Erreichte Punkte:				

Viel Glück!

Aufgabe 1.B: Sortierverfahren **(16 Punkte)**

a) (8 Punkte) Beweisen Sie, dass für die im Folgenden definierte Funktion $f(n)$ die Beziehung $f(n) = O(n^3)$ gilt.

Verwenden Sie für Ihren Beweis die Konstante $c = 2$ und wählen Sie einen geeigneten Wert für n_0. Beachten Sie, dass für einen gültigen Beweis auch die formale Definition angegeben werden muss.

$$f(n) = \begin{cases} n^3 + 50 \cdot n^2, & \text{falls } n > 10 \\ 5 \cdot n^3 + n^2 + 20, & \text{sonst} \end{cases}$$

Kreuzen Sie anschließend in der folgenden Tabelle die zutreffenden Felder für die oben angeführte Funktion $f(n)$ an:

$f(n)$ ist	$\Theta(.)$	$O(.)$	$\Omega(.)$	keines
n^3				
$n^2\sqrt{n}$				
n^4				
$n^2 \log_2(2^n)$				

b) (8 Punkte) Welche in der Vorlesung behandelten Sortierverfahren eignen sich am Besten für die unten angeführten Folgen. Begründen Sie Ihre Antwort.

- Eine sehr große Anzahl an Rechnungen, die jeweils mit einer Jahreszahl und einer 5-stelligen Rechnungsnummer versehen sind, soll sortiert werden.
- Bei einem Transport sollen schwere Pakete aufsteigend nach ihrer Bezeichnung sortiert werden. Schlüsselbewegungen sind dadurch wesentlich aufwendiger als Schlüsselvergleiche. Die Länge der Bezeichnung eines Paketes ist nicht bekannt.
- Eine beliebige Folge soll absteigend sortiert werden.
- In eine bereits absteigend sortierte Folge wird ein beliebig großes Element am Ende angehängt. Die neu entstandene Folge soll nun erneut absteigend sortiert werden.

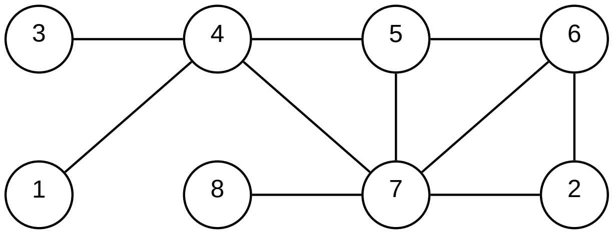
Aufgabe 2.B: Graphen und Optimierung (14 Punkte)

a) (3 Punkte)

Welche aus der Vorlesung bekannten Datenstrukturen und Verbesserungen werden benötigt, damit der Gesamtaufwand des Algorithmus von Kruskal $O(|E| \log |E|)$ nicht übersteigt? Begründen Sie Ihre Antwort!

b) (11 Punkte)

Gegeben ist ein ungerichteter, zusammenhängender Graph $G = (V, E)$:



Gegeben ist ferner folgender Algorithmus:

Algorithmus Was-bin-ich?

Input: Graph $G = (V, E)$
Output: Kantenmenge F und Feld $X[1 \dots |V|]$
$F = \{\}$;
$X[1 \dots |V|] = (-1, -1, \dots, -1)$;
A(4);

Prozedur A(v)

$X[v] = X[v] + 1$;
for all Knoten $w \in N(v)$ in der gegebenen Ordnung **do**
  if $X[w] < 0$ **then**
    $F = F \cup (v, w)$;
    A(w);
  end if
  $X[w] = X[w] + 1$;
end for

- Welche Kanten liefert der Algorithmus für den oben angeführten Graphen in F zurück? Markieren Sie diese Kanten im Graphen. Geben Sie weiters das Feld X nach der Ausführung des Algorithmus an.
- Beschreiben Sie in einem Satz, was dieser Algorithmus allgemein für einen beliebigen ungerichteten, zusammenhängenden Graphen macht und in F bzw. X zurückliefert.
- Welche Laufzeit ergibt sich bei diesem Algorithmus im Worst-Case, in Θ-Notation, in Abhängigkeit der Anzahl der Knoten $|V|$ und Kanten $|E|$, für einen beliebigen ungerichteten, zusammenhängenden Graphen, wenn dieser in Form von Adjazenzlisten gespeichert ist?

Aufgabe 3.B: ADTS und Suchen **(20 Punkte)**

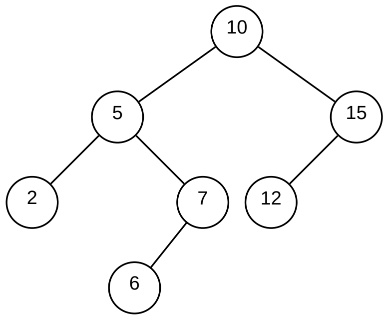
a) (8 Punkte)

- Zeichnen Sie einen AVL Baum, bei dem eine Postorder-Durchmusterung folgendes Resultat liefert:

$$< 10, 20, 35, 55, 50, 40, 30 >$$

Ist der AVL-Baum mit der oben angeführten Postorder-Durchmusterung eindeutig rekonstruierbar? Begründen Sie Ihre Antwort.

- Gegeben sei weiters folgender AVL-Baum:



Löschen Sie den Schlüssel 12 aus dem Baum. Ist nach dem Entfernen des Schlüssels 12 die AVL-Baum Eigenschaft verletzt? Falls ja, führen Sie alle notwendigen Reorganisationsmaßnahmen durch, um erneut einen gültigen AVL-Baum zu erhalten, und zeichnen Sie den resultierenden AVL-Baum.

b) (12 Punkte) Schreiben Sie detaillierten Pseudocode für die Funktion $merge(L_1, L_2)$. Dabei sind die Parameter L_1 und L_2 jeweils Zeiger auf das erste Element einer aufsteigend sortierten, azyklischen, einfach verketteten Liste. Als Rückgabe soll die Funktion $merge(L_1, L_2)$ einen Zeiger auf das erste Element einer aufsteigend sortierten, einfach verketteten Liste liefern, die alle Elemente aus der Liste L_1 und L_2 enthält.