



186.172 Algorithmen und Datenstrukturen 1 VL 4.0

3. Übungstest SS 2008

27. Juni 2008

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:	15	18	17	50
Erreichte Punkte:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Viel Erfolg!

Aufgabe 1.A: $\Omega/O/\Theta$ -Notation**(15 Punkte)**

a) (5 Punkte)

Gegeben sei die folgende Funktion:

$$f(n) = \begin{cases} n \log_2 n^3, & \text{falls } n \text{ eine 2er-Potenz ist} \\ 3n^2 - 2n & \text{sonst} \end{cases}$$

Beweisen oder widerlegen Sie:

$$f(n) = \Theta(n^2)$$

Beachten Sie, dass für einen korrekten Beweis auch geeignete Werte für die Konstanten n_0 , c_1 und c_2 angegeben werden müssen!

b) (6 Punkte)

Seien $f(n)$, $g(n)$ und $h(n)$ Funktionen mit positivem Wertebereich. Sind die folgenden Aussagen wahr oder falsch?

Aussage	wahr	falsch
$f(n) = O(g(n)) \wedge h(n) = \Omega(g(n)) \Rightarrow f(n) = O(h(n))$		
$f(n) = O(n \cdot g(n)) \Rightarrow g(n) = \Omega(n \cdot f(n))$		
$f(n) = O(g(n)) \vee f(n) = \Omega(g(n)) \Rightarrow f(n) = \Theta(g(n))$		

c) (4 Punkte)

Gegeben ist der folgende Algorithmus:

Algorithmus 1: WasBinIch(A, i)**Eingabe:** Feld $A[1, \dots, n]$ mit n ganzen Zahlen, Feldindex i

```

1:  $m = i$ ;
2: für  $j = i + 1, i + 2, \dots, n$  {
3:   falls  $A[j] < A[m]$  dann {
4:      $m = j$ ;
5:   }
6: }
7: Vertausche  $A[i]$  und  $A[m]$ ;
8: falls  $i < n$  dann {
9:   WasBinIch( $A, i + 1$ );
10: }
```

- Um welches konkrete aus der Vorlesung bekannte Verfahren handelt es sich beim Algorithmus **WasBinIch**?
- Geben Sie den Aufwand von **WasBinIch** in Abhängigkeit der Feldgröße n für den Worst- und den Best-Case in Θ -Notation an.

Aufgabe 3.A: Minimale Spannende Bäume (MST)

(17 Punkte)

a) (12 Punkte)

Für den Datentyp der dynamisch disjunkten Mengen (DDM, Union Find) benötigt man ein Array $father$, das jedem Knoten v einen Vorgänger $father[v]$ zuweist, sowie die Prozeduren **makeset**(v), **findset**(v) und **union**(v, w). Um die in der Vorlesung behandelte Verbesserung *Vereinigung nach Höhe* zu realisieren, braucht man zusätzlich ein Array $height$, das zu jedem Knoten v die Höhe $height[v]$ speichert.

Die Prozeduren **makeset**(v) und **findset**(v) sind dabei wie folgt implementiert:

Algorithmus 2: makeset(v)

- 1: $father[v] = v$;
- 2: $height[v] = 0$;

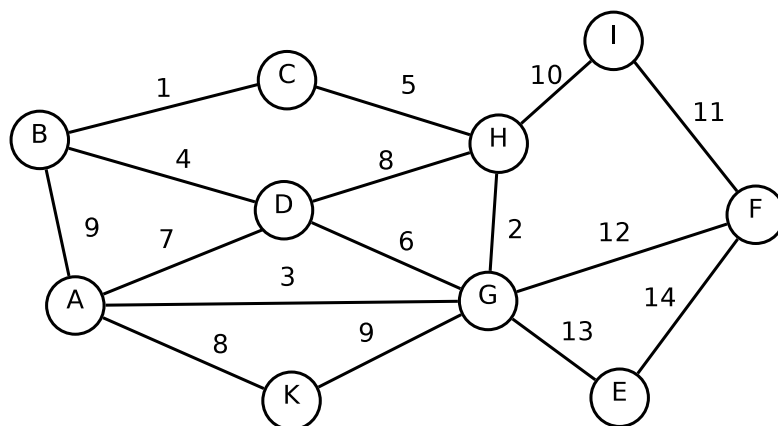
Algorithmus 3: findset(v)

- 1: $h = v$;
- 2: **solange** $father[h] \neq h$ {
- 3: $h = father[h]$;
- 4: }
- 5: **retourniere** h ;

- Schreiben Sie detaillierten Pseudocode für die Prozedur **union**(v, w) (v und w jeweils Repräsentant ihrer Menge), der die Verbesserung *Vereinigung nach Höhe* realisiert.
- Für welchen aus der Vorlesung bekannten MST-Algorithmus ist die DDM-Datenstruktur eine effiziente Möglichkeit, Kreise zu finden?

b) (5 Punkte)

Führen Sie in dem unten abgebildeten Graphen den Algorithmus von *Kruskal* zum Finden eines minimalen Spannbaums durch (die Zahlen bei den Kanten bezeichnen die jeweiligen Kantenkosten). Zeichnen Sie den Spannbaum direkt im Graphen ein und notieren Sie die genaue Reihenfolge, in der die Kanten in den Baum aufgenommen wurden. *Falls* Sie einen Startknoten benötigen, verwenden Sie dazu den Knoten A.





186.172 Algorithmen und Datenstrukturen 1 VL 4.0

3. Übungstest SS 2008

27. Juni 2008

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:	15	18	17	50
Erreichte Punkte:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Viel Glück!

Aufgabe 1.B: $\Omega/O/\Theta$ -Notation**(15 Punkte)**

a) (4 Punkte)

Gegeben ist der folgende Algorithmus:

Algorithmus 1: WasMacheIch(A, i)**Eingabe:** Feld $A[1, \dots, n]$ mit n ganzen Zahlen, Feldindex i

```

1: falls  $i = n$  dann {
2:   Abbruch: fertig!
3: }
4:  $m = i$ ;
5: für  $j = i + 1, i + 2, \dots, n$  {
6:   falls  $A[j] > A[m]$  dann {
7:      $m = j$ ;
8:   }
9: }
10: Vertausche  $A[i]$  und  $A[m]$ ;
11: WasMacheIch( $A, i + 1$ );

```

- Um welches konkrete aus der Vorlesung bekannte Verfahren handelt es sich beim Algorithmus **WasMacheIch**?
- Geben Sie den Aufwand von **WasMacheIch** in Abhängigkeit der Feldgröße n für den Worst- und den Best-Case in Θ -Notation an.

b) (6 Punkte)

Seien $f(n)$, $g(n)$ und $h(n)$ Funktionen mit positivem Wertebereich. Sind die folgenden Aussagen wahr oder falsch?

Aussage	wahr	falsch
$f(n) = O(g(n)/n) \Rightarrow g(n) = \Omega(f(n)/n)$		
$g(n) = \Omega(f(n)) \wedge g(n) = O(h(n)) \Rightarrow h(n) = \Omega(f(n))$		
$(f(n) = O(g(n)) \wedge f(n) = \Omega(g(n))) \vee f(n) \neq \Theta(g(n))$		

c) (5 Punkte)

Gegeben sei die folgende Funktion:

$$f(n) = \begin{cases} n^2 \log_3 n^2, & \text{falls } n \text{ eine 3er-Potenz ist} \\ 2n^3 - 3n & \text{sonst} \end{cases}$$

Beweisen oder widerlegen Sie:

$$f(n) = \Theta(n^3)$$

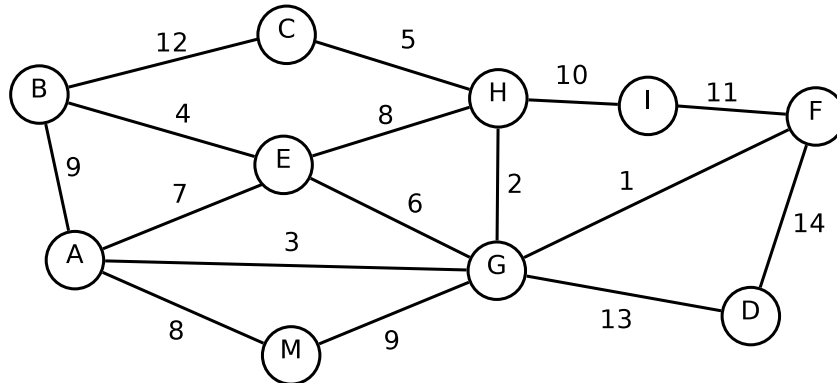
Beachten Sie, dass für einen korrekten Beweis auch geeignete Werte für die Konstanten n_0, c_1 und c_2 angegeben werden müssen!

Aufgabe 3.B: Minimale Spannende Bäume (MST)

(17 Punkte)

a) (5 Punkte)

Führen Sie in dem unten abgebildeten Graphen den Algorithmus von *Prim* zum Finden eines minimalen Spannbaums durch (die Zahlen bei den Kanten bezeichnen die jeweiligen Kantenkosten). Zeichnen Sie den Spannbaum direkt im Graphen ein und notieren Sie die genaue Reihenfolge, in der die Kanten in den Baum aufgenommen wurden. Falls Sie einen Startknoten benötigen, verwenden Sie dazu den Knoten A.



b) (12 Punkte)

Für den Datentyp der dynamisch disjunkten Mengen (DDM, Union Find) benötigt man ein Array *father*, das jedem Knoten v einen Vorgänger $father[v]$ zuweist, sowie die Prozeduren **makeset**(v), **findset**(v) und **union**(v, w).

Die Prozeduren **makeset**(v) und **union**(v, w) sind dabei wie folgt implementiert:

Algorithmus 2: makeset(v)

1: $father[v] = v;$

Algorithmus 3: union(v, w)

// u und v jeweils

// Repräsentant ihrer Menge

1: $father[v] = w;$

- Schreiben Sie detaillierten Pseudocode für die Prozedur **findset**(v), der die *Methode der Pfadverkürzung* realisiert.
- Für welchen aus der Vorlesung bekannten MST-Algorithmus ist die DDM-Datenstruktur eine effiziente Möglichkeit, Kreise zu finden?