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

186.172 Algorithmen und Datenstrukturen 1 VL 4.0

2. Übungstest SS 2009

09. Juni 2009

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:	18	16	16	50
Erreichte Punkte:				

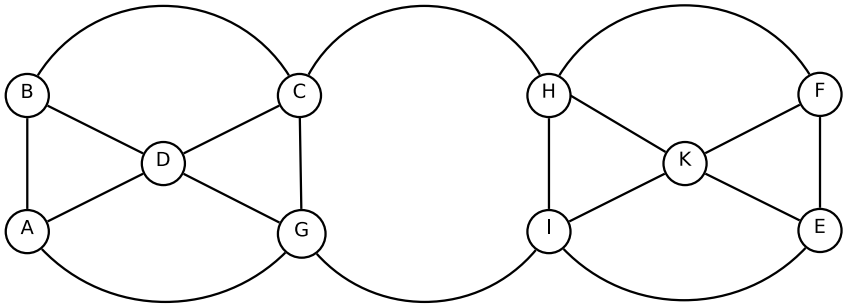
Viel Erfolg!

Aufgabe 1.A: Graphen **(18 Punkte)**

a) (12 Punkte) Ein ungerichteter Graph $G = (V, E)$ heißt *bipartit*, wenn man die gesamte Knotenmenge V in zwei disjunkte Untermengen U und W so aufspalten kann, dass für alle Kanten $(u, w) \in E$ gilt: $u \in U$ und $w \in W$.

Schreiben Sie einen möglichst effizienten Algorithmus, der berechnet, ob ein Graph G bipartit ist. Geben Sie weiters die Laufzeit ihres Algorithmus in Abhängigkeit der Anzahl der Knoten n im Worst-Case an.

b) (6 Punkte) Auf dem gegebenen Graphen wird die aus dem Skriptum bekannte **Tiefensuche** durchgeführt. Welche der folgenden Listen von besuchten Knoten können dabei in genau dieser Reihenfolge entstehen. *Hinweis:* Die Nachbarn eines Knotens können in beliebiger Reihenfolge abgearbeitet werden.



Reihenfolge: A B C D E F G H I K ☐ ja ☐ nein

Reihenfolge: A B D C G H I K F E ☐ ja ☐ nein

Reihenfolge: B A D C G H I K F E ☐ ja ☐ nein

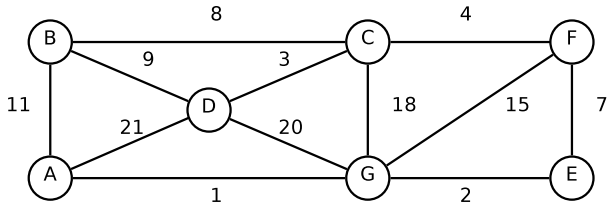
Reihenfolge: A B C D H K F E I H ☐ ja ☐ nein

Reihenfolge: K E F H I G C D A B ☐ ja ☐ nein

Reihenfolge: C G I K E F H B D A ☐ ja ☐ nein

Aufgabe 2.A: Optimierung **(16 Punkte)**

Gegeben ist der folgende gewichtete ungerichtete Graph G:



a) (6 Punkte) Führen Sie in dem Graphen G 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 B.

b) (10 Punkte) Gegeben Sei folgender Algorithmus *WasBinIch*, der auf einen Graphen $G(V, E)$ angewendet wird und zusätzlich zwei Knoten $s, t \in V$ als Parameter erhält.

Algorithmus 1 WasBinIch$(G(V, E), s, t)$

```
1: fuer alle v ∈ V : d[v] = ∞;
2: d[s] = 0; Q = V;
3: solange Q nicht leer {
4:    Entnimm jenes u aus Q mit minimalem d[u];
5:    falls d[u] = ∞ dann Ausgabe: Fehler & Abbruch des Algorithmus;
6:    falls u = t dann Ausgabe: d[t] & Abbruch des Algorithmus;
7:    für alle e = (u, v) mit v ∈ N(u) {
8:       falls d[v] > d[u] + w_e dann {
9:          d[v] = d[u] + w_e;
10:      }
11:   }
12: }
```

- Beschreiben Sie kurz, was der Algorithmus *WasBinIch* in einem Graphen berechnet und auf welchem aus der Vorlesung bekannten Prinzip dieser Algorithmus beruht.
- Wenden Sie diesen Algorithmus auf den oben angeführten Graphen G an (die Kantenbeschriftungen entsprechen dem Kantengewicht w_e zwischen den verbundenen Knoten). Geben Sie die Ausgabe des Algorithmus an, wenn dieser durch den Aufruf von WasBinIch $(G(V, E), D, A)$ gestartet wird. Tragen Sie weiters den Zustand des Feldes d nach der Ausführung des Algorithmus in folgender Tabelle ein:

$v \in V$	A	B	C	D	E	F	G
$d[v]$							

Aufgabe 3.A: Hash und Suchverfahren **(16 Punkte)**

a) (8 Punkte) Gegeben ist ein Feld, das die folgenden Strings in der angegebenen Reihenfolge enthält:

$$\langle algodat,\ alt,\ hans,\ haselnuss,\ heinz,\ herbert,\ opa,\ petzi\rangle.$$

Führen Sie in diesem Feld eine Binärsuche nach dem String *petzi* durch. Geben Sie dabei in jedem Schritt die jeweiligen Bereichsgrenzen an. Das erste Element des Feldes hat Index 1, falls notwendig wird bei Indexberechnungen abgerundet.

Geben Sie die Laufzeit des Worst-Case für das Suchen eines Strings mit Binärsuche innerhalb einer sortierte Folge in Θ-Notation an, und zwar in Abhängigkeit der Anzahl der Elemente n und der maximalen Stringlänge k.

b) (8 Punkte) Gegeben ist eine Hashtabelle mit Tabellengröße $m = 7$, die zur Kollisionsbehandlung Double Hashing einsetzt.

$$h_1(k) = k \bmod 7$$
$$h_2(k) = (k+1) \bmod 7$$

- Fügen Sie den Wert 14 in die folgende Tabelle ein:

0	1	2	3	4	5	6
21			24		12	

- Fügen Sie den Wert 12 in die folgende Tabelle ein:

0	1	2	3	4	5	6
7			3	11	5	

- Welches Problem kann beim Einfügen in die Hashtabelle bei der Verwendung der oben angegebenen Hashfunktionen auftreten? Wie müssen die Funktionen $h_1(k)$ und/oder $h_2(k)$ abgeändert werden, um dieses Problem zu beheben?

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

186.172 Algorithmen und Datenstrukturen 1 VL 4.0

2. Übungstest SS 2009

09. Juni 2009

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	18	16	50
Erreichte Punkte:				

Viel Glück!

Aufgabe 1.B: Hash und Suchverfahren **(16 Punkte)**

a) (8 Punkte) Gegeben ist eine Hashtabelle mit Tabellengröße $m = 7$, die zur Kollisionsbehandlung Double Hashing einsetzt.

$$h_1(k) = k \bmod 7$$
$$h_2(k) = (k + 1) \bmod 7$$

- Fügen Sie den Wert 19 in die folgende Tabelle ein:

0	1	2	3	4	5	6
14			10	11	5	

- Fügen Sie den Wert 21 in die folgende Tabelle ein:

0	1	2	3	4	5	6
14			24		12	

- Welches Problem kann beim Einfügen in die Hashtabelle bei der Verwendung der oben angegebenen Hashfunktionen auftreten? Wie müssen die Funktionen $h_1(k)$ und/oder $h_2(k)$ abgeändert werden, um dieses Problem zu beheben?

b) (8 Punkte) Gegeben ist ein Feld, das die folgenden Strings in der angegebenen Reihenfolge enthält:

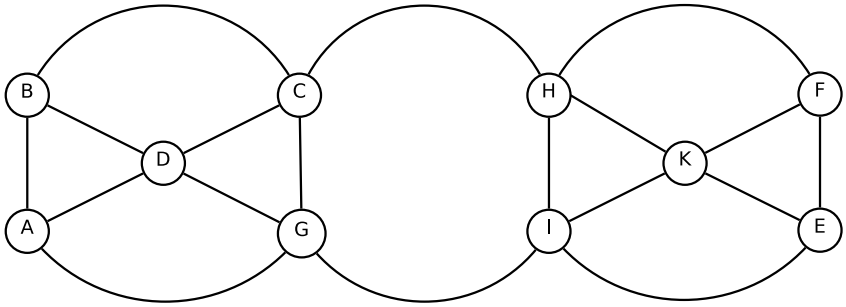
$$\langle \textit{algorithmus}, \textit{alt}, \textit{hannes}, \textit{hans}, \textit{heinz}, \textit{herbert}, \textit{opa}, \textit{peter} \rangle.$$

Führen Sie in diesem Feld eine Binärsuche nach dem String *peter* durch. Geben Sie dabei in jedem Schritt die jeweiligen Bereichsgrenzen an. Das erste Element des Feldes hat Index 1, falls notwendig wird bei Indexberechnungen abgerundet.

Geben Sie die Laufzeit des Worst-Case für das Suchen eines Strings mit Binärsuche innerhalb einer sortierte Folge in Θ-Notation an, und zwar in Abhängigkeit der Anzahl der Elemente n und der maximalen Stringlänge k.

Aufgabe 2.B: Graphen **(18 Punkte)**

a) (6 Punkte) Auf dem gegebenen Graphen wird die aus dem Skriptum bekannte **Tiefensuche** durchgeführt. Welche der folgenden Listen von besuchten Knoten können dabei in genau dieser Reihenfolge entstehen. *Hinweis:* Die Nachbarn eines Knotens können in beliebiger Reihenfolge abgearbeitet werden.



Reihenfolge: A B C D E F G H I K ☐ ja ☐ nein

Reihenfolge: A B D C G H I K F E ☐ ja ☐ nein

Reihenfolge: K E F H I G C D A B ☐ ja ☐ nein

Reihenfolge: C G I K E F H B D A ☐ ja ☐ nein

Reihenfolge: A B C D H K F E I H ☐ ja ☐ nein

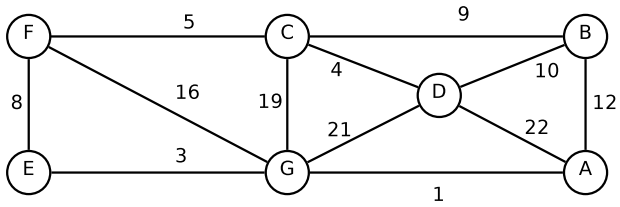
Reihenfolge: B A D C G H I K F E ☐ ja ☐ nein

b) (12 Punkte) Ein ungerichteter Graph $G = (V, E)$ heißt *bipartit*, wenn man die gesamte Knotenmenge V in zwei disjunkte Untermengen U und W so aufspalten kann, dass für alle Kanten $(u, w) \in E$ gilt: $u \in U$ und $w \in W$.

Schreiben Sie einen möglichst effizienten Algorithmus, der berechnet, ob ein Graph G bipartit ist. Geben Sie weiters die Laufzeit ihres Algorithmus in Abhängigkeit der Anzahl der Knoten n im Worst-Case an.

Aufgabe 3.B: Optimierung **(16 Punkte)**

Gegeben ist der folgende gewichtete ungerichtete Graph G:



a) (6 Punkte) Führen Sie in dem Graphen G 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 B.

b) (10 Punkte) Gegeben Sei folgender Algorithmus *WasBinIch*, der auf einen Graphen $G(V, E)$ angewendet wird und zusätzlich zwei Knoten $s, t \in V$ als Parameter erhält.

Algorithmus 2 WasBinIch$(G(V, E), s, t)$

```
1: fuer alle v ∈ V : d[v] = ∞;
2: d[s] = 0; Q = V;
3: solange Q nicht leer {
4:    Entnimm jenes u aus Q mit minimalem d[u];
5:    falls d[u] = ∞ dann Ausgabe: Fehler & Abbruch des Algorithmus;
6:    falls u = t dann Ausgabe: d[t] & Abbruch des Algorithmus;
7:    für alle e = (u, v) mit v ∈ N(u) {
8:       falls d[v] > d[u] + w_e dann {
9:          d[v] = d[u] + w_e;
10:      }
11:   }
12: }
```

- Beschreiben Sie kurz, was der Algorithmus *WasBinIch* in einem Graphen berechnet und auf welchem aus der Vorlesung bekannten Prinzip dieser Algorithmus beruht.
- Wenden Sie diesen Algorithmus auf den oben angeführten Graphen G an (die Kantenbeschriftungen entsprechen dem Kantengewicht w_e zwischen den verbundenen Knoten). Geben Sie die Ausgabe des Algorithmus an, wenn dieser durch den Aufruf von WasBinIch $(G(V, E), D, A)$ gestartet wird. Tragen Sie weiters den Zustand des Feldes d nach der Ausführung des Algorithmus in folgender Tabelle ein:

$v \in V$	A	B	C	D	E	F	G
$d[v]$							