Algorithmische Mathematik

Wintersemester 2013
Prof. Dr. Marc Alexander Schweitzer und
Dr. Einar Smith
Patrick Diehl und Daniel Wissel

universitätbonn

Übungsblatt 10: „Weihnachtsblatt“.

Abgabe am **06.01.2014**.

Hinweise:

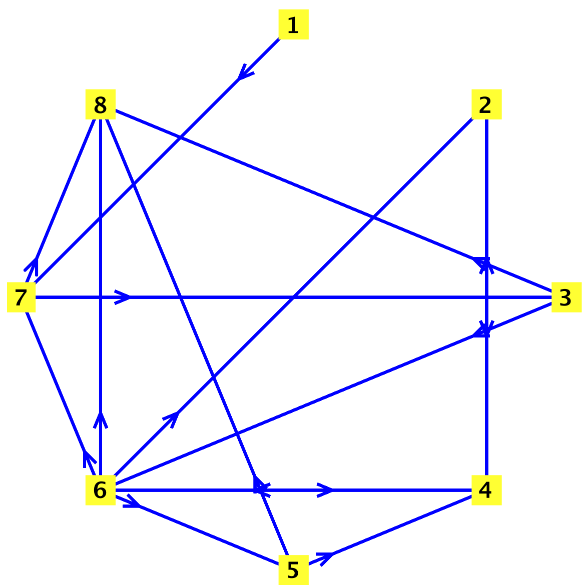
a. Es steht Ihnen frei, dieses Übungsblatt zu bearbeiten.

b. Es wird Ihnen allerdings empfohlen, dieses Übungsblatt als Vorbereitung auf die Klausur am 13.02.2014 zu bearbeiten.

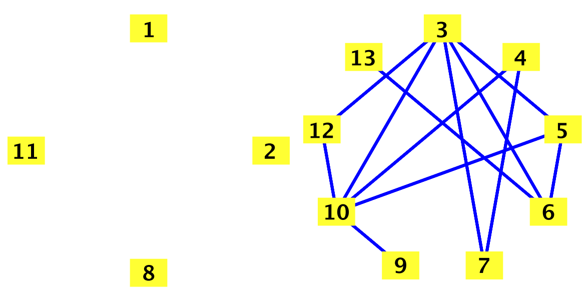
c. Mit den Punkten dieses Übungsblatts können Sie eines der Blätter 2 – 9 ausgleichen.

Aufgabe 1. (Zusammenhangskomponenten)

a. Bestimmen Sie in diesem Graph alle starken Zusammenhangskomponenten.



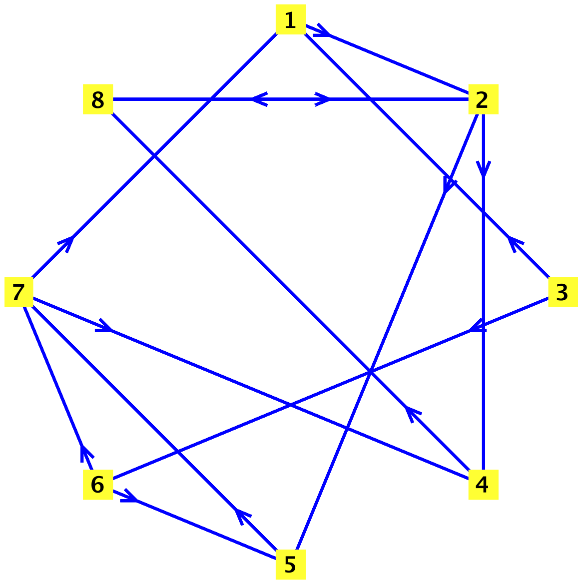
b. Bestimmen Sie in diesem Graph alle Zusammenhangskomponenten.



(1 + 1 = 2 Punkte)

Aufgabe 2. (Breitensuche)

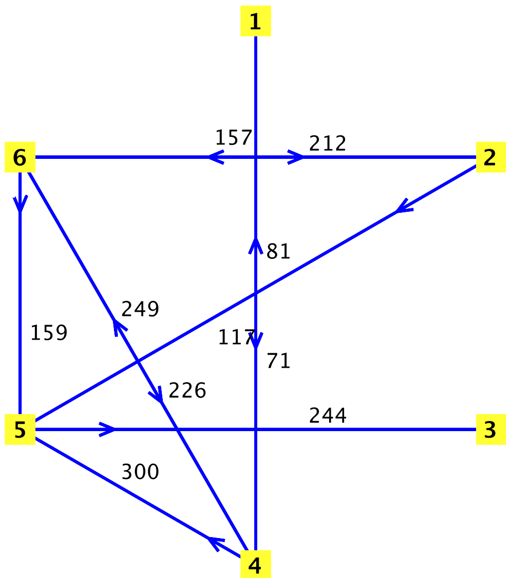
Wenden Sie den Algorithmus für die Breitensuche 2.25 auf den Graphen an, um zu entscheiden, ob der Weg $\pi : 3 \to 8$ existiert. Geben Sie in jedem Schritt die Mengen R, S, T an.



(3 Punkte)

Aufgabe 3. (Dijkstras-Algorithmus)

Wenden Sie den Algorithmus von Dijkstra auf den Graphen an, um den kürzesten Weg $\pi : 1 \to 6$ zu finden. Geben Sie in jedem Schritt die Tabelle aus der Illustration in der Vorlesung an.



(3 Punkte)

Aufgabe 4. (Programmierung auf dem Papier)

Schreiben Sie eine Funtion `float median(int n, int x[])`, die in einem Array den Median findet. Der Median kann wie folgt bestimmt werden:

a. Die Zahlen werden absteigend sortiert.

b. Wenn die Anzahl der Werte ungerade ist, ist die mittlere Zahl der Median.

c. Wenn die Anzahl der Werte gerade ist, ist der Median das arithmetische Mittel der beiden mittleren Zahlen.

(2 Punkte)

Aufgabe 5. (Zyklische Graphen)

Zeigen Sie, dass ein Graph $G = (V, E)$ mit $grad(v) \geq 2$ für alle $v \in V$ einen Zyklus enthält.

(2 Punkte)

Aufgabe 6. (Die Türme von Hanoi)

Bei diesem Spiel gibt es drei Stäbe A, B und C. Auf den Stab A werden x verschieden große Scheiben der Größe nach gestapelt. Ziel ist es diese Scheiben der Größe nach auf den Stab C um zu sortieren. Folgender rekursiver Algorithmus beschreibt das Umsortieren der Scheiben.

```
funktion bewege (Zahl i, Stab a, Stab b, Stab c) {
    falls (i > 0) {
        bewege(i-1, a, c, b);
        verschiebe oberste Scheibe von a nach c;
        bewege(i-1, b, a, c);
    }
}
```

Geben Sie die Komplexität des Algorithmus in der Landaunotation $\mathcal{O}$ an. Beginnen Sie mit $t_n = t_{n-1} + 1 + t_{n-1}$ und wenden dann die Rekursion auf t_{n-1} an.

(3 Punkte)

Aufgabe 7. (Gleitkommazahlen)

Wir betrachten einen Ein-Byte-Rechner mit Gleitkomma-Arithmetik. Bei der (normalisierten) Zahldarstellung werden ein Bit für das Vorzeichen s, vier Bits für die Mantisse m $(1 \leq m < 2)$ und drei Bits für den Exponenten e bei einem Bias von 4 verwendet. Die führende Eins in der Mantisse wird nicht abgespeichert. Somit haben Zahlen im Rechner die Form

$$z = (-1)^s \cdot m \cdot 2^{e-4} \text{ mit } m = 1 + \sum_{i=1}^{4} m_i \cdot 2^{-i} \text{ und } e = \sum_{j=1}^{3} e_j \cdot 2^{j-1}$$

a. Welche Darstellung haben die Zahlen 6.5 und -0.375?

b. Geben Sie die maximal und minimal darstellbaren Zahlen $z_{\max}$ und $z_{\min}$ sowie die betragsmäßig kleinste darstellbare Zahl $z_{|\min|}$ an.

c. Auch hier werden nicht darstellbare Zahlen auf die nächste darstellbare Zahl gerundet. Bestimmen Sie den maximalen absoluten und relativen Rundungsfehler für reelle Zahlen im Bereich $[z_{|\min|}, z_{\max}]$.

$(1 + 1 + 1 = 3$ Punkte$)$

Aufgabe 8. (Zahldarstellung)

a. Schreiben Sie die Dezimalzahl $(100101)_{10}$ als Hexadezimalzahl.

b. Schreiben Sie die Hexadezimalzahl $(100101)_{16}$ als Binärzahl.

c. Seien z_1 und z_2 zwei natürliche Zahlen mit identischer Ziffernfolge $d_{N-1}d_{N-2}\dots d_0$ bezüglich unterschiedlicher Basen b_1 und b_2. Welche der folgenden Aussagen sind wahr?

- falls $b_1 > b_2$, so ist $z_1 > z_2$
- falls $z_1 > z_2$, so ist $b_1 > b_2$
- falls b_1 b_2 teilt, so teilt z_1 z_2
- falls z_1 z_2 teilt, so teilt b_1 b_2
- $z_1 + z_2$ besitzt in der Basis $b_1 + b_2$ die selbe Ziffernfolge wie z_1 bzw. z_2
- $z_1 \cdot z_2$ besitzt in der Basis $b_1 + b_2$ die selbe Ziffernfolge wie z_1 bzw. z_2

($\frac{1}{2} + \frac{1}{2} + 3 = 4$ Punkte)

Programmieraufgabe 1. (Binärer Suchbaum)

Auf dem vierten Übungsblatt wurde der Begriff des Binärbaums eingeführt. Unser Binärbaum speichert in jedem Knoten einen Integer-Wert. Ein Binärbaum lässt sich zu einem binären Suchbaum erweitern, wenn gilt, dass der Inhalt des linken Kindes immer $\leq$ dem Inhalt des Elternknoten und der Inhalt des rechten Kindes immer $>$ dem Inhalt des Elternknoten ist. Implementieren Sie ein `struct node` für einen Knoten eines binären Suchbaums und folgende Methoden:

a. `void insert(**node root, int element)`

b. `int depth(*node root)`

c. `int contains(*node root, int element)`

d. `void delete(*node root)`

Implementieren Sie zusätzlich noch folgende Methode. Was macht diese Methode?

```
print(Knoten) {

    if Knoten.links !=null then
        print(knoten.links)
    endif

    printf(knoten.wert)

    if knoten.rechts != null then
        print(knoten.rechts)
    endif
}
```

Schreiben Sie für alle Operationen sinnvolle Testfälle in Ihr Hauptprogramm. Anhand dieser Testfälle muss ersichtlich sein, dass die Operationen auf Ihrer Datenstruktur korrekt funktionieren.

$(2 + 2 + 2 + 2 = 8$ Punkte)

Abgabe am 06.01.2014 zwischen Vorlesung A und Vorlesung B

Programmieraufgabe 2. (Flächeninhalt eines Dreiecks)

Gegeben sei ein Dreieck mit Seitenlängen a, b, c. Nach der Formel von Heron von Alexandria kann der Flächeninhalt des Dreiecks über die Formel

$$\triangle_{\text{Heron}} = \sqrt{s(s-a)(s-b)(s-c)} \quad \text{wobei} \quad s := \frac{1}{2}(a+b+c)$$

berechnet werden.

a. Implementieren Sie eine Funktion `float areaHeron(float in[])`, welche die Fläche nach der Formel von Heron berechnet. In dem Array `in` stehen die Längen der Seiten a, b und c. Geben Sie die berechnete Fläche für die Seitenlängen, die in der Datei *input1.dat* stehen, in der Konsole aus.

b. Was passiert, wenn Sie die Seitenlängen vor der Eingabe wie folgt $a \geq b \geq c$ sortieren und folgende Formel verwenden:

$$\triangle^*_{\text{Heron}} = \frac{1}{4}\sqrt{(a+(b+c))(c-(a-b))(c+(a-b))(a+(b-c))} \;?$$

Mehr Informationen zu diesem Thema findet man beispielsweise in

> W. Kahan: *Miscalculating Area and Angles of a Needle-like Triangle*, from Lecture Notes for Introductory Numerical Analysis Classes.

Hinweis: Achten Sie auf die korrekte Klammerung der Ausdrücke um die gewünschten Effekte zu sehen.

(2 + 2 = 4 Punkte)

Abgabe am 06.01.2014 zwischen Vorlesung A und Vorlesung B

Programmieraufgabe 3. (Implementierung von Mengen (2))

Erweitern Sie Ihren Code zur Programmieraufgabe 2, Blatt 4 um folgende Operationen:

a. Vereinigungsmenge $M_1 \cup M_2 := \{x | (x \in M_1) \vee (x \in M_2)\}$
`void setunion(struct node* s1, struct node* s2, struct node** result)`

b. Teilmenge $M_1 \subseteq M_2$: `int subset(struct node* s1, struct node* s2)`

c. Gleichheit $M_1 = M_2$: `int equal(struct node* s1, struct node* s2)`

d. Sortieren der Elemente (<): `void sort(struct node* s)`

Schreiben Sie für alle Operationen sinnvolle Testfälle in Ihr Hauptprogramm. Anhand dieser Testfälle muss ersichtlich sein, dass die Operationen auf Ihrer Datenstruktur korrekt funktionieren.

(3 + 1 + 1 + 3 = 8 Punkte)

Abgabe am 06.01.2014 zwischen Vorlesung A und Vorlesung B

Wir wünschen Ihnen frohe Weihnachten und einen guten Rutsch ins Neue Jahr!
Vi ønsker dere en god jul og et godt nytt år!