



Algorithmische Mathematik I

Wintersemester 2015 / 2016
Prof. Dr. Sven Beuchler
Markus Siebenmorgen



Aufgabenblatt zur Wiederholung: Teil 1.

Abgabedatum: -.

- Aufgabe 1.** a) Was ist eine B -adische Darstellung mit fixer Länge ω für natürliche Zahlen?
b) Man stelle die Zahlen $x = 21$ und $y = 14$ in einer 2-adischen Darstellung mit $\omega = 5$ dar.
c) Man stelle die Zahl $x + y$ in einer 2-adischen Darstellung mit $\omega = 5$ dar. Was tritt auf?

Aufgabe 2. Es sei $B, t \in \mathbb{N}$, $B > 1$. Was ist eine normalisierte Gleitkommazahl zur Basis B mit Mantissenlänge t ? Es sei $G_{10,3,-5,5}$ ein Gleitkommasystem.

- a) Man berechne $rd(x)$ für $x = 0.2176$.
b) Man berechne $x \tilde{+} y$ für $x = 0.218$ und $y = -1.74$.

Aufgabe 3. Es sei $x \in \mathbb{R}$ ein exakter Wert und \tilde{x} die gestörten Daten. Definieren Sie absoluten und relativen Fehler. Weiterhin sei $\phi : \mathbb{R} \mapsto \mathbb{R}$ gegeben durch

$$\phi(x) = \sqrt{1 + x^2}.$$

Wie pflanzt sich der relative Fehler für $\tilde{x} \rightarrow x$ von $\phi(\tilde{x})$ gegenüber $\phi(x)$ fort?

Aufgabe 4. Es seien die Zahlen

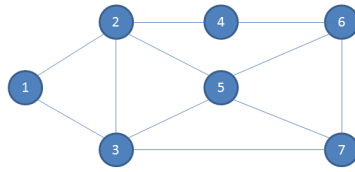
7 9 3 2 12 8 4 1

zu sortieren. Demonstrieren Sie die folgenden Verfahren anhand obigen Beispiels.

- a) Mergesort,
b) Quicksort,
c) Heapsort,
d) Bubblesort.

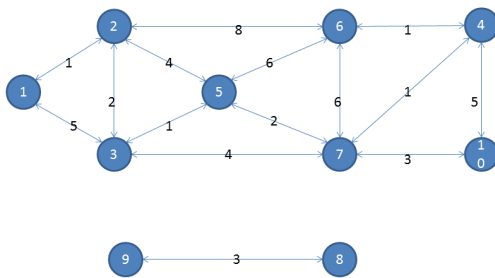
Geben Sie jeweils möglichst gute Abschätzungen für die mittlere Laufzeit und den Speicherplatz der entsprechenden Verfahren an.

Aufgabe 5. Es sei $G = (V, E)$ ein einfacher ungerichteter Graph. Man gebe 6 äquivalente Definitionen, daß G ein Baum ist.



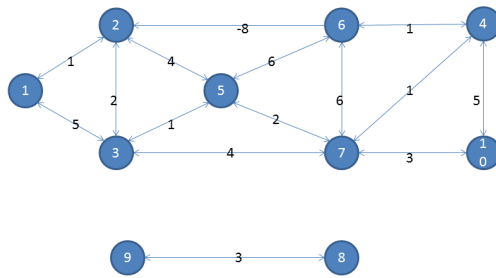
Man demonstriere Tiefen- und Breitensuche anhand des obigen Graphen mit Startknoten 1.

Aufgabe 6. Wir betrachten den folgenden gewichteten Graphen:



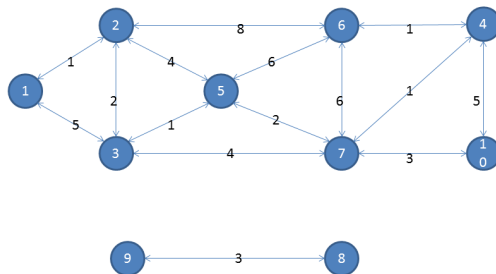
- Man berechne mit dem Algorithmus von Dijkstra die kürzesten Weglängen zu allen Knoten von Knoten 1 und gebe für den Knoten 4 den kürzesten Weg an.
- Welche Voraussetzungen müssen für den Algorithmus von Dijkstra an die Gewichtsfunktion gestellt werden?
- Welche Komplexität besitzt der Algorithmus von Dijkstra?

Aufgabe 7. Wir betrachten den folgenden gewichteten, gerichteten Graphen:



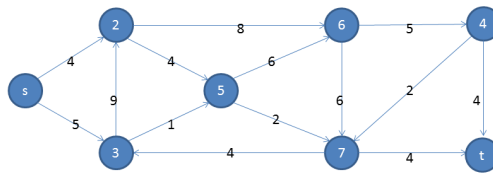
- Kann man den Algorithmus von Dijkstra zur Berechnung kürzester Wege auf diesen Graphen anwenden?
- Enthält der Graph negative Zyklen?
- Man berechne die kürzesten Weglängen zu allen Knoten von Knoten 1 und gebe für den Knoten 3 den längsten Weg an.
- Welche Komplexität besitzt der Algorithmus von Floyd-Warshall?

Aufgabe 8. Wir betrachten den folgenden gewichteten Graphen:



Man bestimme mit dem Algorithmus von Kruskal einen minimal spannenden Baum in allen Zusammenhangskomponenten von G .

Aufgabe 9. Wir betrachten das folgende Netzwerk $N = (G, c, s, t)$.



- Man definiere die Begriffe Netzwerk, $s-t$ -Fluß in einem Netzwerk, $s-t$ -Schnitt, minimaler $s-t$ -Schnitt.
- Wie lautet das max flow min cut Theorem?
- Man berechne mit dem Edmond-Karps-Algorithmus den maximalen $s-t$ -Fluß in obigem Netzwerk.

Aufgabe 10. Es sei $G = (V, E)$ ein einfacher, ungerichteter Graph.

- Was ist der Grad einer Ecke?
- Man zeige die folgende Aussage: Durch Orientierung aller Kanten in G kann G genau dann zu einem gerichteten Graphen \tilde{G} mit $|\delta^+(v)| = |\delta^-(v)|$ für alle $v \in V$ transformiert werden, wenn alle Ecken geraden Grad haben.

Aufgabe 11. Es sei $G = (V, E)$ ein einfacher, ungerichteter Graph.

- Was ist ein Kreis in G ?
- Man zeige, daß G durch Orientierung aller Kanten in G zu einem gerichteten Graphen \tilde{G} transformiert werden kann, sodaß \tilde{G} keinen Kreis besitzt.

Aufgabe 12. Es sei $G = (V, E)$ ein einfacher ungerichteter Graph.

- Wann heißt G bipartit? Was ist ein Matching M ?
- G heißt k -regulär, falls $|\delta(v)| = k$ für alle $v \in V$. Man zeige, daß jeder bipartite k -reguläre Graph ein perfektes Matching besitzt, falls $k \geq 1$.

Aufgabe 13. (Code-Verständnis)

Notieren Sie die Bildschirmausgabe, die nachfolgendes Code-Fragment erzeugt:

```
for (int i=0; i<10; i++)
{
    if ((i%2)==0)
        printf("%d\n", i);
}
```

Aufgabe 14. (Code-Verständnis)

Notieren Sie die Bildschirmausgabe, die nachfolgendes Code-Fragment erzeugt:

```
for (int i=10; i>0; i--)
{
    int j = i * 2;
    if (j > 5)
        printf("%d\n", j);
}
```

Aufgabe 15. (Code-Verständnis)

Notieren Sie die Bildschirmausgabe, die nachfolgendes Code-Fragment erzeugt:

```
for (int i=0; i<4; i++)
{
    for (int j=0; j<4; j++)
    {
        if (((i+j)%2)==0)
            printf("(%d,%d)\n", i, j);
    }
}
```

Programmieraufgabe 1. (Spannweite einer Zahlenfolge)

In der Statistik interessiert man sich gelegentlich für die *Spannweite* einer gegebenen Folge x_i , $i = 0, \dots, n - 1$ von n Zahlen. Die Spannweite S ist hierbei als die Differenz vom größten und kleinsten Wert gegeben:

$$S = \max_i x_i - \min_i x_i$$

Schreiben Sie eine Funktion `double spannweite(double* x, int n)`, die zu einem Array x der Länge n die Spannweite berechnet und als Rückgabewert liefert.

Hinweis:

Im Rahmen dieser Aufgabe darf Pseudo-Code verwendet werden. Unter Pseudo-Code verstehen wir hierbei C/C++-artigen Code, bei dem insbesondere Konstrukte wie `if`-Abfragen, Schleifen, Zuweisungen und arithmetischen Operationen verwendet werden sollen. Auf den Array-Eintrag i eines Arrays `double* x` wird entweder über die mathematische Schreibweise x_i oder die C-Schreibweise `x[i]` zugegriffen. Das erste Element eines Arrays x ist `x[0]`.

Der Einsatz der `pow/exp`-Funktion sowie die Verwendung von anderen in C definierten oder über Header-Dateien einbinbaren Mathematik-Funktionen ist **nicht** gestattet. Ebenso sind **keine** Mengenschreibweisen oder mathematische Schreibweisen (z.B. \sum , \prod , $!$) gestattet. Selbstverständlich darf auch reines C oder C++ als Programmiersprache verwendet werden. Kleinere syntaktische Fehler führen hier nicht zum Punktabzug, solange der eigentliche Algorithmus verständlich ist.