Logik und Komplexität
Aufgabensammlung

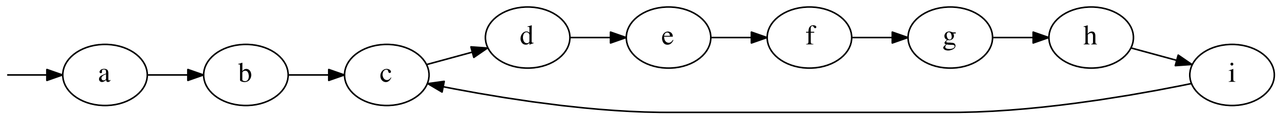
Jan Hladik & Stephan Schulz

2015

1 Übungen

Aufgabe 1 (3+3P)

a) Simulieren Sie den Hase/Igel-Algorithmus auf der folgenden zyklischen Liste. Schreiben Sie zu jedem Schritt auf, wo sich Hase und Igel jeweils befinden. Nach wie vielen schritten treffen sich Hase und Igel?



Lösung:

Schritt	Hase	Igel	Schritt	Hase	Igel
Start	a	a	10.		
1.	c	b	11.		
2.	e	c	12.		
3.	g	d	13.		
4.	i	e	14.		
5.	d	f	15.		
6.	f	g	16.		
7.	**h**	**h**	17.		
8.			18.		
9.			19.		

Hase und Igel treffen sich nach 7 Schritten

b) Entwerfen Sie eine zyklische Liste mit 6 Knoten, die den Worst-Case für Hase/Igel demonstriert. Nach wie vielen Schritten treffen sich Hase und Igel in Ihrem Beispiel?

Lösung: Worst-Case: Hase ist bei Eintritt der Schleife direkt vor Igel. Also: Schleife vom letzten zum ersten Knoten, damit 6 Schritten notwendig.

Aufgabe 2 (3+3)

Betrachten Sie die Funktion $f(x) = \sin(x)^2$

a) Zeigen Sie: $f \in O(1)$

b) Können Sie den Nachweis mit dem Grenzwertkritierum und der Regel von l'Hôpital führen? Warum oder warum nicht?

Lösung:

a) : Anwendung der Definition:
Für eine Funktion f bezeichnet $\mathcal{O}(f)$ die Menge aller Funktionen g mit

$$\exists k \in \mathbb{N} \quad \exists c \in \mathbb{R}^{\geq 0} \quad \forall n > k : g(n) \leq c \cdot f(n)$$

Also: Sei $k = 0, c = 2$. Es gilt: $\sin(x)^2 \in [0,1]$ für alle $x \in \mathbb{N}$. Also gilt insbesondere:

$$\forall n > \underbrace{k}_{=0} : \underbrace{\sin(n)^2}_{\leq 1} \leq \underbrace{\underbrace{c}_{=2} \cdot 1}_{=2}$$

Damit ist $\sin(x) \in \mathcal{O}(1)$

b) Nein, da $\frac{\sin^2(x)}{1}$ nicht konvergiert. Das Grenzwert-Kriterium ist hinreichend, aber nicht notwendig.

Aufgabe 3 (3+3)

a) Zeigen Sie: $x \log_2 x \in O(x^2)$

b) Zeigen Sie: $\log_2 x \in O(x^2)$

Lösung:
Vorbemerkung: $\log(x) = \frac{\ln(x)}{\ln(2)}$. Damit $\log_2(x)' = \frac{1}{\ln(2)} \ln(x)' = \frac{1}{\ln(2)} \frac{1}{x} = c\frac{1}{x}$ für geeignetes c.

a) Behauptung: $x \log_2 x \in O(x^2)$
Wir zeigen per Grenzwertkriterium: $\lim_{x\to\infty} \frac{x \log_2 x}{x^2} \in \mathbb{R}^{\geq 0}$.

$$\begin{aligned}
\lim_{x\to\infty} \frac{x \log_2 x}{x^2} &= \lim_{x\to\infty} \frac{\log_2 x}{x} \text{ (kürzen von } x\text{)} \\
&= \lim_{x\to\infty} \frac{\frac{1}{\ln(2)x}}{1} \text{ (l'Hôpital)} \\
&= \lim_{x\to\infty} \frac{1}{\ln(2)x} \text{ (rechnen)} \\
&= 0
\end{aligned}$$

Also: Der Grenzwert existiert und ist in $\mathbb{R}^{\geq 0}$. Damit gilt die Behauptung.

b) Behauptung: $\log_2 x \in O(x^2)$
Wir zeigen per Grenzwertkriterium: $\lim_{x\to\infty} \frac{\log_2 x}{x^2} \in \mathbb{R}^{\geq 0}$.

$$\begin{aligned}
\lim_{x\to\infty} \frac{\log_2 x}{x^2} &= \lim_{x\to\infty} \frac{\frac{1}{\ln(2)x}}{2x} \text{ (l'Hôpital)} \\
&= \lim_{x\to\infty} \frac{1}{\ln(2) \cdot 2x} \text{ (rechnen)} \\
&= 0
\end{aligned}$$

Also: Der Grenzwert existiert und ist in $\mathbb{R}^{\geq 0}$. Damit gilt die Behauptung.

Aufgabe 4 (1+1+1+6 Punkte)
Betrachten Sie die folgende C-Funktion:

```
int machwas(int n)
{
    int k = n;
    sum = 0;
    for(i=0; i< n; i++)
    {
        for(j=0; j<k; j++)
        {
            sum++;
        }
        k = k/2;
    }
    return sum;
}
```

a) Bestimmmen Sie den Rückgabwert für die Eingaben $n = 4, n = 8, n = 12$.

b) Bestimmen Sie das kleinste $k \in \mathbb{N}$ so dass die Laufzeitkomplexität von `machwas()` in $\mathcal{O}(n^k)$ ist.

Aufgabe 5 (5 Punkte)
Sortieren Sie die Folge S mit dem Bubble-Sort-Algorithmus. Geben Sie hierzu den Zustand von S nach jedem Durchlauf der Hauptschleife an.

$$S = \{3, 8, 17, 5, 15, 2, 1, 12, 4, 9\}$$

Wieviele Vertauschungen benötigen Sie?

Aufgabe 6 (6 Punkte)
Sortieren Sie die Folge S mit dem in der Vorlesung bespriebenen LL-Quicksort-Algorithmus. Verwenden Sie dabei jeweils das *letzte* Element einer (Teil-) Folge als Pivot.

Geben Sie hierzu jeweils an, wo sich die Pointer befinden, wenn eine Vertauschung von Elementen stattfindet, und was das Resultat der Vertauschung ist.

$$S = \{3, 8, 17, 5, 15, 2, 1, 12, 4, 9\}$$

Wieviele Vertauschungen benötigen Sie?

Aufgabe 7
6+1P Bestimmen die die Komplexität ($\mathcal{O}$-Notation) des (hypothetischen) *Perfect-Quicksort-Algorithmus*, der im folgenden Punkt vom Standard-Quicksort Algorithmus abweicht: Perfect QS findet immer das perfekte Pivot-Element, aber benötigt hierfür bei n Elementen zusätzliche Zeit der Größenordnung $\mathcal{O}(n^2)$.

Würden Sie diesen Algorithmus gegenüber dem normalen Quicksort bevorzugen?

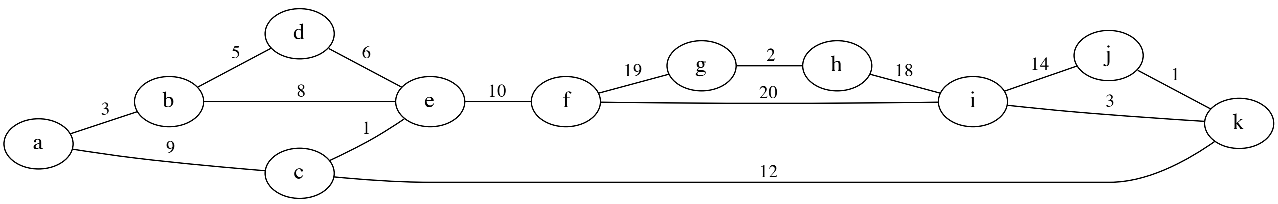
LÖsung:

Für *Perfect Quicksort* ergibt sich

$$P(n) = 2 \cdot P(\frac{n}{2}) + k_1 \cdot n + k_2 n^2$$

, wobei $k_1 n$ die Partitionierung repräsentiert, $k_2 n^2$ die Auswahl des Pivot. Mit dem Master-Theorem haben wir $a = 2, b = 2, d = 2$, also $a < b^d$ (Fall 1) und $P \in \mathcal{O}(n^2)$. *Perfect Quicksort* ist damit in der selben Klasse wie Bubblesort oder Selectionsort, und im Durchschnittsfall schlechter als normales Quicksort.

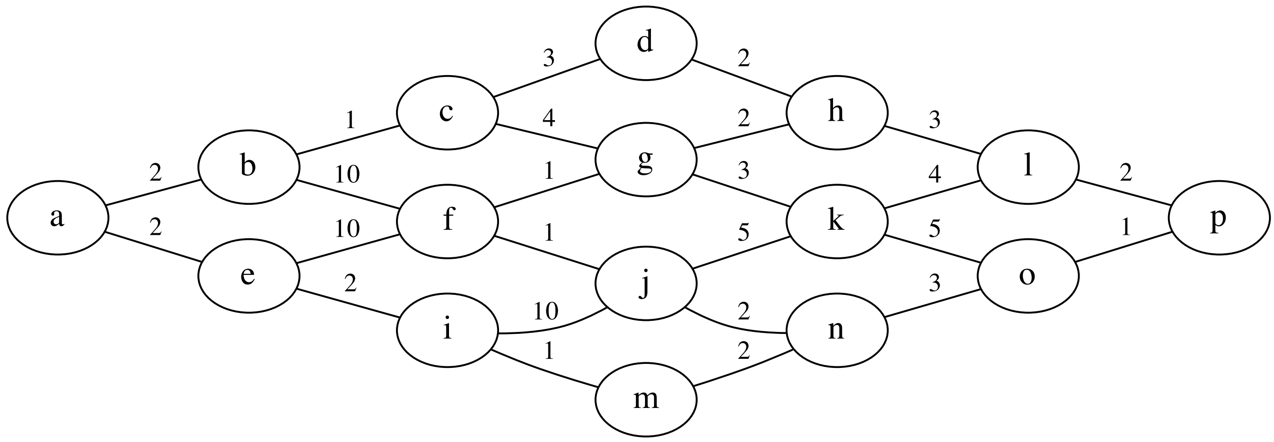
Aufgabe 8 (3+2+2P)
Gegeben sei der Graph G:



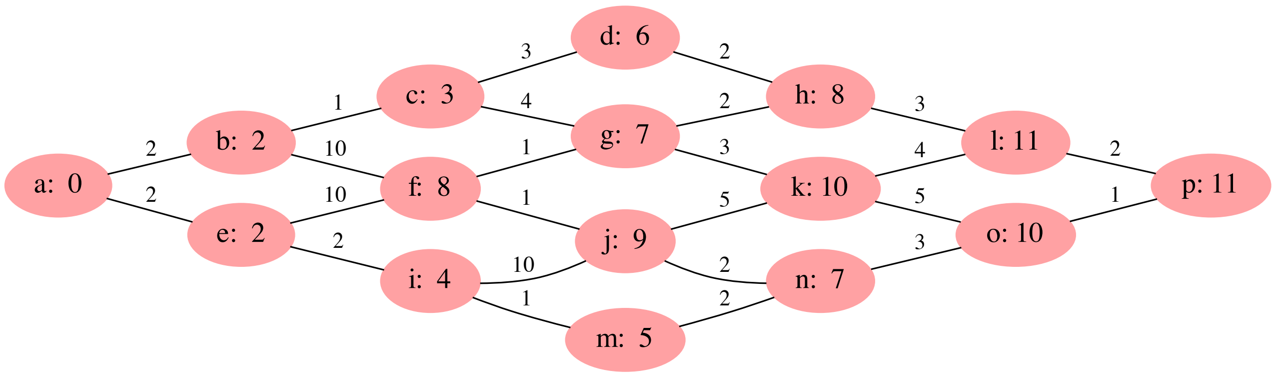
- Bestimmen Sie für G einen minimalen Spannbaum mit Hilfe des Prim-Algorithmus.

- Wie viele Kanten enthält der MSB für einen Graphen mit n Kanten? Begründen Sie Ihre Antwort.
- Begründen Sie anhand von G, warum es zur Bestimmung des MSB eines Graphen im Allgemeinen nicht ausreicht, die günstigsten Kanten zu verwenden.

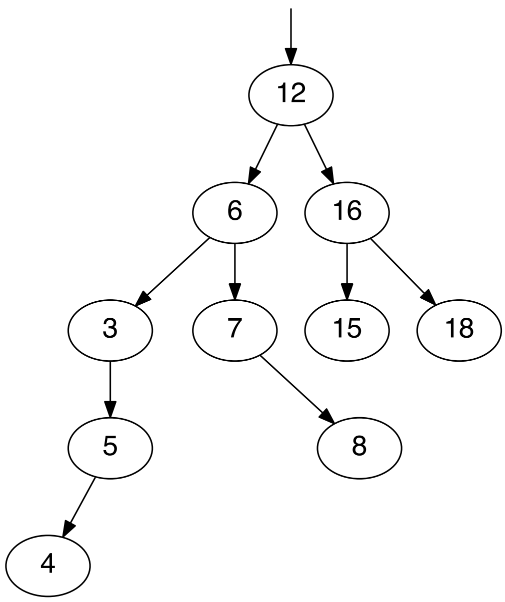
Aufgabe 9 (6 Punkte)
Gegeben Sei der Graph H.



Bestimmen Sie für alle Knoten die minimale Entfernung zum Knoten a mit Hilfe des Dijkstra-Algorithmus.
Lösung:



Aufgabe 10 (3+3+3P)
Gegeben sei der binäre Suchbaum B.



a) Fügen Sie zu B Knoten für die Schlüssel 9, 14 und 8 hinzu.

b) Entfernen Sie aus B (Achtung: dem ursprünglichen Baum, nicht dem Resultat von Aufgabenteil a) die Knoten 3, 12 und 7.

c) Bestimmen Sie eine möglichst kleine Anzahl von Rotationen, durch die B die Bedingung erfüllt, dass für jeden Knoten der Höhenunterschied der beiden Kinder maximal 1 beträgt.