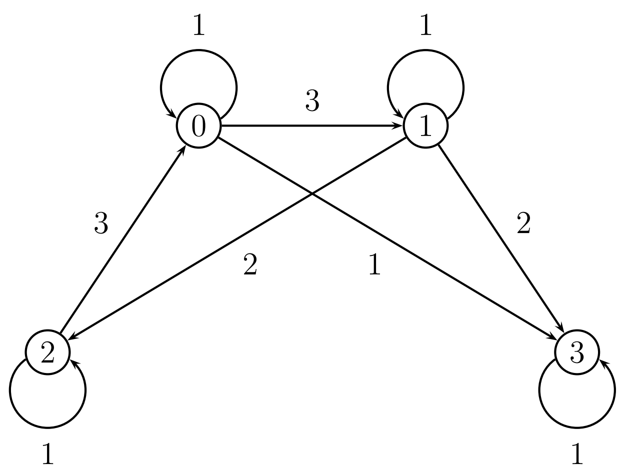
Grundbegriffe der Informatik
Musterlösung zu Aufgabenblatt 9

Aufgabe 9.1 (5+2 Punkte)

Für Graphen mit gewichteten Kanten steht in der Adjazenzmatrix an der Stelle i, j eine 0, falls es keine Kante von i nach j gibt, und das Gewicht der Kante (i, j) sonst.
Der Warshall-Algorithmus wird für solche Graphen genauso durchgeführt wie für ungewichtete Graphen in der Vorlesung angegeben.

a) Führen Sie für folgenden Graph mit gewichteten Kanten den Warshall-Algorithmus durch; geben Sie dabei nur die Matrix W an, die sich nach Abschluss der Initialisierung ergeben hat, sowie die Matrizen W_0, W_1, W_2, W_3, die sich jeweils nach dem ersten, zweiten, dritten und vierten Durchlauf der äußeren Schleife beim zweiten Teil des Algorithmus ergeben.



b) Welche Bedeutung hat die Zahl, die am Ende in der resultierenden Matrix an der Stelle i, j steht?

Lösung 9.1

a) $W = \begin{pmatrix} 1 & 3 & 0 & 1 \\ 0 & 1 & 2 & 2 \\ 3 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ $W_0 = \begin{pmatrix} 1 & 3 & 0 & 1 \\ 0 & 1 & 2 & 2 \\ 3 & 3 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ $W_1 = \begin{pmatrix} 1 & 3 & 2 & 2 \\ 0 & 1 & 2 & 2 \\ 3 & 3 & 2 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

$$W_2 = \begin{pmatrix} 2 & 3 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 2 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad W_3 = \begin{pmatrix} 2 & 3 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 2 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

b) Falls $i \neq j$ gilt, steht an der Stelle i, j eine 0, falls es keinen Pfad von i nach j gibt, und sonst die größte Zahl w_{ij}, so dass es einen Pfad von i nach j gibt, auf dem jede Kante mindestens das Gewicht w_{ij} hat.

   Falls $i = j$ gilt, steht an der Stelle i, j eine 1, falls es keinen Pfad der Länge > 0 von i nach i gibt, und sonst die größte Zahl w_{ij}, so dass es einen Pfad von i nach i gibt, auf dem jede Kante mindestens das Gewicht w_{ij} hat.

   Hinweis: Dieser Unterschied für den Fall $i = j$ ist etwas unschön und unintuitiv; falls jemand nur gesehen hat, was der Eintrag besagt wenn $i \neq j$ gilt, ist das die volle Punktzahl wert.

Aufgabe 9.2 (2+2+2 Punkte)
Beweisen oder widerlegen Sie:

a) $n! \in \Omega(n^2)$

b) $\sqrt[2]{n} \in O(\sqrt[3]{n})$

c) Für alle Funktionen $f(n), g(n), h(n), i(n) > 0$ gilt:
   $f(n) \in O(h(n)) \wedge g(n) \in O(i(n)) \Rightarrow f(g(n)) \in O(h(i(n)))$.

Lösung 9.2

a) Die Aussage ist wahr:
   Für $n \geq 2$ gilt $n! = n \cdot (n-1) \cdot (n-2)! \geq n \cdot \frac{n}{2} \cdot 1 = \frac{1}{2}n^2$.
   Für $n_0 = 2$ und $c = \frac{1}{2}$ gilt also: $\forall n \geq n_0 : n! \geq cn^2$, und es folgt $n! \in \Omega(n^2)$.

b) Die Aussage ist falsch:

   Seien $n_0 \in \mathbb{N}_0$ und $c > 0$ beliebig aber fest gewählt. Wir wählen $n \geq \max(n_0 + 1, c^6 + 1)$ und erhalten $\sqrt[2]{n}/\sqrt[3]{n} = n^{1/2}/n^{1/3} = n^{\frac{1}{2}-\frac{1}{3}} = n^{1/6} > c$, und somit folgt $\sqrt[2]{n} > c\sqrt[3]{n}$.

   Also gibt es für jedes $n_0 \in \mathbb{N}_0$ und jedes $c > 0$ ein $n > n_0$, so dass nicht $\sqrt[2]{n} \leq c\sqrt[3]{n}$ gilt. Dies bedeutet, dass $\sqrt[2]{n} \notin O(\sqrt[3]{n})$ gilt.

c) Die Aussage ist falsch:

Wir wählen $f(n) = g(n) = \begin{cases} 1 & \text{falls } n = 0 \\ 1/n & \text{sonst} \end{cases}$

und $h(n) = i(n) = 1$.

Da $f(n)$ und $g(n)$ jeweils durch 1 nach oben beschränkt sind, gilt $f(n) \in O(h(n))$ und $g(n) \in O(i(n))$.

Es gilt jedoch $f(g(n)) = \begin{cases} 1 & \text{falls } n = 0 \\ n & \text{sonst} \end{cases}$,

während $h(i(n)) = 1$ gilt.

Da nach der Vorlesung $n \notin O(1)$ gilt, folgt, dass die Aussage falsch ist.

Aufgabe 9.3 (2+3 Punkte)
a sei ein Array der Länge n und k eine natürliche Zahl, für die $0 < k < n$ gilt. Anfangs enthalte das Array r nur Nullen.
Gegeben sei folgender Algorithmus:

```
for j ← k − 1 to n − 1 do
    for i ← j − k + 1 to j do
    r[j] ← r[j] + a[i]
    od
    r[j] ← r[j]/k
od
```

und außerdem der Algorithmus:

for $i \leftarrow 0$ **to** $k-1$ **do**
    $r[k-1] \leftarrow r[k-1] + a[i]$
od
for $j \leftarrow k$ **to** $n-1$ **do**
    $r[j] \leftarrow (a[j] - a[j-k] + r[j-1])$
od
for $j \leftarrow 0$ **to** $n-1$ **do**
    $r[j] \leftarrow r[j]/k$
od

a) Was berechnen die beiden Algorithmen im Array r?

b) Welche der beiden Algorithmen besitzt die kürzere Laufzeit? Begründen Sie Ihre Antwort.

Lösung 9.3

a) Im Array r steht am Ende an der j-ten Stelle das arithmetische Mittel (der Durchschnitt) der k vorhergehenden Einträge in $a[j-k+1]$ bis $a[j]$.

b) Algorithmus 1: Die äußere Schleife wird $(n-k+1)$ mal durchlaufen, die innere Schleife k mal, was insgesamt zu $(n-k+1)\cdot k+(n-k+1) = (n-k+1)(k+1)$ Operationen führt.

   Algorithmus 2: Die Schleifen werden nacheinander k mal, $(n-k)$ mal und n mal durchlaufen, was insgesamt zu $2n$ Operationen führt.

   Es ist intuitiv klar, dass der zweite Algorithmus meistens schneller ist als der erste, da die Summe nicht jedes Mal neu berechnet wird, sondern der vorherige Wert aktualisiert wird. Wir rechnen aus, für welche n genau das der Fall ist, für welche n also gilt:

   $(n-k+1)(k+1) > 2n$

$$
\begin{aligned}
& (n-k+1)(k+1) > 2n \\
\iff\ & n(k+1) - (k-1)(k+1) > 2n \\
\iff\ & n(k-1) > (k-1)(k+1) \\
\iff\ & k > 1 \wedge n > k+1
\end{aligned}
$$

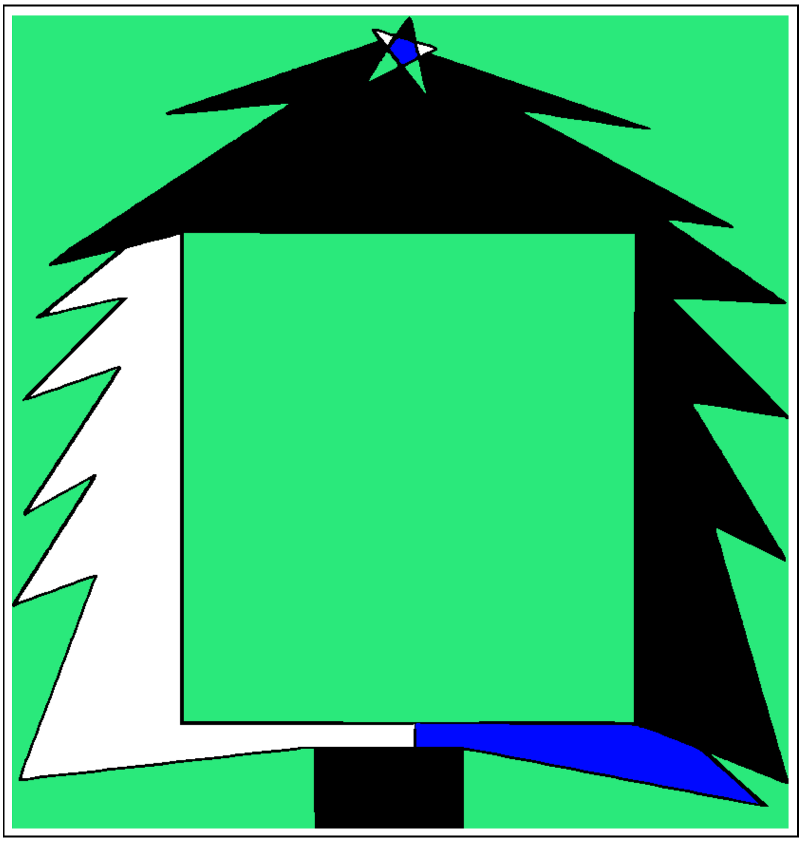
Der einzige Fall, für den der zweite Algorithmus nicht schneller als der erste ist, ist also, wenn $n = k + 1$ oder $k = 1$ gilt, in welchen Fällen beide Algorithmen gleich schnell sind.

Hinweis: Das ist hier sehr ausführlich. Punkte gibt es für das Ausrechnen/Abschätzen der Laufzeiten des Algorithmus und einen richtigen Vergleich durch Abschätzen, das auch weniger präzise als hier sein darf.

Aufgabe 9.4 (2 Punkte)

Färben Sie die Flächen der folgenden Abbildung mit möglichst wenig Farben, so dass 2 adjazente Flächen nie die gleiche Farbe haben.

Lösung 9.4



Hinweis: Wenn man die Aufgabenstellung so interpretiert, dass zwei Flächen adjazent sind, falls sie einen gemeinsamen Punkt haben (statt einer gemeinsamen Kante), benötigt man 5 Farben. Dafür gibt es auch volle Punktzahl