Universität Bremen

FB3: Mathematik/Informatik

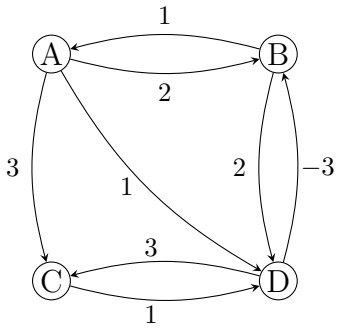
Prof. Dr. Daniel Neuen
Jens Schlöter

Wintersemester 2023/2024

Algorithmentheorie

Übungsblatt 5 (Abgabe am 08.01.2024, 12:15 Uhr)

Übung 5.1 (6 Punkte)



Führt den Bellman-Ford-Algorithmus auf obigem Graphen mit Startknoten A aus. Die Kanten werden dabei in lexikographischer Reihenfolge betrachtet, d.h. $(A, B), (A, C), (A, D), (B, A), (B, D), (C, D), (D, B), (D, C)$. Ihr dürft den Algorithmus vorzeitig abbrechen, wenn sich keine weiteren Änderungen mehr ergeben. Gebt für jede Runde des Algorithmus die Updates der Distanzlabel an.

Übung 5.2 (8 Punkte)

Ihr seid Teamleiter bei einem IT-Dienstleister. Jede Woche kann euer Team genau ein Aufgabenpaket erledigen. Ihr unterscheidet dabei zwischen Routineaufgaben, die ohne große Vorbereitung erledigt werden können, und komplexen Aufgaben. Eure Aufgabe ist es, für alle Wochen $1, \ldots, n$ zu entscheiden, ob Ihr das Routine-Aufgabenpaket, das komplexe Aufgabenpaket, oder gar kein Aufgabenpaket bearbeitet. Durch das Bearbeiten des Routine-Aufgabenpakets in Woche $i \in \{1, \ldots, n\}$ erzielt ihr Einnahmen in Höhe r_i. Durch das Bearbeiten des komplexen Aufgabenpakets in Woche $i \in \{1, \ldots, n\}$ erzielt ihr Einnahmen in Höhe c_i. Das komplexe Aufgabenpaket benötigt allerdings eine intensive Vorbereitung, so dass ihr, wenn ihr in Woche i das komplexe Aufgabenpaket bearbeitet, in Woche $i-1$ kein Aufgabenpaket bearbeiten könnt. Das bedeutet auch, dass ihr in Woche 1 kein komplexes Aufgabenpaket bearbeiten könnt. Wenn ihr in einer Woche kein Aufgabenpaket bearbeitet, erzielt ihr in dieser Woche keine Einnahmen. Ihr kennt also die Werte $r_1, \ldots, r_n$ sowie $c_1, \ldots, c_n$ und eure Aufgabe ist es, die Aufgabenpakete für die Wochen $1, \ldots, n$ so auszuwählen, dass maximale Einnahmen erzielt werden.

(a) Zeigt, dass der folgende Algorithmus nicht optimal ist, indem Ihr eine Instanz angebt, für die der Algorithmus eine falsche Antwort zurückgibt. Gebt für diese Instanz die optimale

Lösung an sowie die Lösung, welche der Algorithmus zurückgibt. Der Algorithmus nimmt an, dass $r_i = c_i = 0$ für alle $i > n$.

Algorithmus 1 : Greedy Algorithmus

```
1 i := 1;
2 while i ≤ n do
3     if c_{i+1} > r_i + r_{i+1} then
4         print("Kein Aufgabenpaket in Woche i");
5         print("Komplexes Aufgabenpaket in Woche i+1");
6         i := i + 2;
7     else
8         print("Routine-Aufgabenpaket in Woche i");
9         i := i + 1;
```

(b) Gebt einen Algorithmus an, der in Polynomialzeit mit Hilfe von dynamischer Programmierung die maximal erzielbaren Einnahmen berechnet. Gebt insbesondere auch die induktive Definition der DP-Tabelle an. Begründet, warum Euer Algorithmus korrekt ist und die gewünschte Laufzeit erreicht.

(c) Erklärt, wie aus der DP-Tabelle eures Algorithmus der Aufgabenplan berechnet werden kann, welcher das maximale Einkommen erzielt.

Übung 5.3 (6 Punkte)

Gegeben sei eine Menge A mit positiven natürlichen Zahlen. Das Ziel ist es, eine Menge $S \subseteq A$ zu finden, so dass $\max\left\{\sum_{a\in S} a, \sum_{a\in A\setminus S} a\right\}$ minimiert wird. Gebt ein dynamisches Programm an, welches das gegebene Problem in Laufzeit $\mathcal{O}(n \cdot \sum_{a\in A} a)$ löst (wobei $n = |A|$). Gebt dabei insbesondere auch die induktive Definition der DP-Tabelle an. Begründet warum euer Algorithmus die optimale Lösung findet und die gewünschte Laufzeit erzielt.