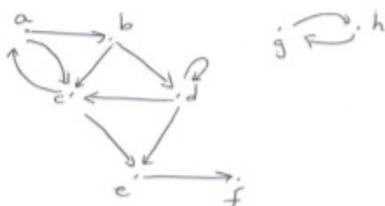


1. Aufgabenblatt für die Vorlesung „Komplexitätstheorie“

Aufgabe 1: 10%

Verwende Algorithmus 2 für die Erreichbarkeit in Graphen aus Kapitel 1 der Vorlesung, um zu entscheiden, ob in folgendem Graphen h von d erreichbar ist. Benutze dieselbe Notation wie im Beispiel aus der Vorlesung.



Aufgabe 2: 15%

Das Rucksackproblem ist als Optimierungsproblem wie folgt formuliert, wobei alle Zahlen *binär* kodiert sind:

Gegeben:

- eine Menge $A = \{a_1, \dots, a_n\}$ von Gegenständen, und für jeden Gegenstand a_i ein Gewicht $g_i \in \mathbb{N}$ und einen Nutzen $n_i \in \mathbb{N}$ und
- eine Gewichtsgrenze $G \in \mathbb{N}$ für einen zu packenden Rucksack.

Ausgabe: berechne eine nutzenmaximale Rucksackfüllung d.h. eine Teilmenge $R \subseteq M$ so dass $\sum_{a_i \in R} g_i \leq G$ und $\sum_{a_i \in R} n_i$ maximal ist.

Berechne die Lösung für folgende Eingabe:

$$A = \{a_1, a_2, a_3, a_4, a_5\} \quad (n_1, n_2, n_3, n_4, n_5) = (3, 8, 3, 6, 2) \quad (g_1, g_2, g_3, g_4, g_5) = (3, 6, 2, 4, 3) \quad G = 8.$$

Argumentiere, dass die Lösung wirklich optimal ist.

Aufgabe 3: 25%

In der Variante als Berechnungsproblem ist beim Rucksackproblem zusätzlich ein Zielnutzen N gegeben, der mindestens erreicht werden muß. Ausgegeben wird eine Rucksackfüllung, die diesen Zielnutzen realisiert.

Zeige: wenn das Berechnungsproblem in Polynomialzeit lösbar ist, dann auch das Optimierungsproblem. (Hinweis: es ist nicht möglich, alle Werte von 0 bis N durchzugehen, da N in binär gegeben ist.)

Aufgabe 4: 25%

In der Variante als Berechnungsproblem ist beim Rucksackproblem ebenfalls ein Zielnutzen N gegeben und die Frage ist, ob eine Rucksackfüllung *existiert*, die diesen Zielnutzen realisiert. Die Ausgabe ist (wie bei jedem Entscheidungsproblem) "ja" oder "nein".

Zeige: wenn das Entscheidungsproblem in Polynomialzeit lösbar ist, dann auch das Berechnungsproblem.

Aufgabe 5: 25%

Entwickle eine (deterministische) Turingmaschine, die als Eingabe $\$bin(n)$ erhält, wobei $bin(n)$ die binäre Kodierung der Zahl n ist, und die diese dann inkrementiert. Es wird angenommen, dass das höchstwertige Bit ganz links steht und das niederwertigste ganz rechts. Beim Start steht der Kopf der Maschine auf dem Symbol $\$$. Dieses darf beim Inkrementieren wenn nötig überschrieben werden.

Gib die Übergänge in graphischer Form an (wie in der Vorlesung). Erkläre die Konstruktion. Gib die Berechnung der TM auf der Eingabe $\$111$ an.

Aufgabe 6: 25% (Zusatzaufgabe)

Betrachte den Algorithmus b -clique aus Kapitel 1 der Vorlesung. Zeige,

- durch Angeben eines Gegenbeispiels, dass das Eliminieren adjazenter Knoten notwendig ist, weil der Algorithmus sonst nicht korrekt ist;
- die Korrektheit des Algorithmus: wenn G eine k -Clique enthält, dann gibt b -clique(G, k) eine k -Clique in G aus.

Hinweis: Für (b) verwende Induktion über k .