Abgabe: 22.11.2016, 12.30 Uhr

Prof. Dr. Heiko Röglin
Clemens Rösner
Institut für Informatik

Übungsblatt 5

Aufgabe 5.1: (3+5 Punkte)

(a) Wir betrachten die Instanz des Rucksackproblems mit 4 Objekten $i = 1, \dots, 4$, die durch die folgende Tabelle gegeben ist.

i	1	2	3	4
p_i	3	1	2	5
w_i	2	4	3	6

Stellen Sie die Tabelle mit den Werten $P(i, w)$, die durch den Algorithmus DynKP berechnet werden, auf und geben Sie eine optimale Lösung für $t = 7$ und $t = 10$ an.

(b) Wir haben gesehen, wie wir mit DynKP optimale Lösungen für das Rucksackproblems mit einer Laufzeit von $O(n^2 W)$ berechnen können, wobei n die Anzahl der Objekte und $W = \max_{i \in \{1,\dots,n\}} w_i$ das maximale Gewicht eines Objektes bezeichnet. Zeigen Sie, wie eine optimale Lösung des Rucksackproblems mit einer Laufzeit von $O(n^2 P)$ berechnet werden kann, wobei $P = \max_{i \in \{1,\dots,n\}} p_i$ den größten Nutzen eines Objektes bezeichnet.

Aufgabe 5.2: (8 Punkte)

Wir betrachten das sogenannte *Partitionsproblem*: Gegeben seien natürliche Zahlen $a_1, \dots, a_n$. Es soll entschieden werden, ob es eine Indexmenge $I \subseteq \{1, \dots, n\}$ gibt, sodass $\sum_{i \in I} a_i = \sum_{i \in \{1,\dots,n\} \setminus I} a_i$ gilt, und gegebenenfalls eine solche Indexmenge zu finden.

Benutzen sie dynamische Programmierung, um das Partitionsproblem mit einer Laufzeit von $O(nS)$ zu lösen, wobei $S = \sum_{i=1}^{n} a_i$ ist.

Hinweis: Eine Indexmenge I besitzt genau dann die gewünschte Eigenschaft, wenn $\sum_{i \in I} a_i = \frac{S}{2}$ gilt.

Aufgabe 5.3: (8 Punkte)

Wir wollen die beiden Algorithmen DynKP und ApproxKP auf einer Menge mit n Objekten vergleichen. In der Vorlesung haben wir bereits gesehen, dass DynKP eine optimale Lösung liefert und ApproxKP eine Lösung liefert, die mindestens halb so gut ist wie eine optimale Lösung. Wir wollen nun untersuchen, wie gut die von ApproxKP gelieferten Lösungen auf zufälligen Instanzen sind.

Erzeugen Sie für $n = 2, \dots, 100$ jeweils 100-mal hintereinander je ein Feld p mit n zufälligen Einträgen für die Nutzenwerte und ein anderes Feld w mit n zufälligen Einträgen für die Gewichte. Dabei sei jeder Eintrag eine zufällige natürliche Zahl aus $\{1, \dots, 100\}$. Führen Sie für eine zufällig gewählte Kapazität

$$t \in \left\{ \max_{i \in \{1,\dots,n\}} w_i, \dots, \sum_{i \in \{1,\dots,n\}} w_i \right\}$$

sowohl DynKP als auch ApproxKP auf diesen Feldern aus. Bestimmen Sie für jedes n die minimale, die maximale und die durchschnittliche Approximationsgüte der Lösung von ApproxKP und stellen Sie diese als Graphen in Abhängigkeit von n dar. Die Approximationsgüte ist der Quotient aus dem optimalen Nutzen und dem Nutzen der Lösung, die ApproxKP berechnet.

Aufgabe 5.4: (6 Zusatzpunkte)

Wir betrachten das folgende Problem. Gegeben sei ein Bild B mit m Zeilen. Jede Zeile ist eine Zeichenfolge aus den Buchstaben R und G. Jedes Zeichen entspricht einem Feld in dem Bild, das entweder grün oder rot gefärbt werden soll, je nachdem, welcher Buchstabe an der entsprechenden Stelle steht. Wir haben die Möglichkeit, horizontale Streifen der Höhe 1 mit einer Farbrolle zu färben, wobei die so erzeugten Streifen jeweils einfarbig sind und kein bereits gefärbtes Feld überstrichen werden darf.

Als Beispiel betrachten wir das folgende Bild B:

$$B = \begin{bmatrix} \mathbf{R} & \mathbf{R} & \mathbf{R} & \mathbf{R} & \mathbf{R} & \mathbf{R} & \mathbf{R} & \mathbf{R} & \mathbf{R} \\ \mathbf{G} & \mathbf{G} & \mathbf{G} & \mathbf{G} & \mathbf{G} & \mathbf{G} & \mathbf{G} & \mathbf{G} & \mathbf{G} \\ \mathbf{G} & \mathbf{G} & \mathbf{R} & \mathbf{R} & \mathbf{R} & \mathbf{G} & \mathbf{G} & \mathbf{G} & \mathbf{G} \end{bmatrix}$$

Das Bild B kann mit 5 Streifen korrekt gefärbt werden: Wir benötigen je einen Streifen für die ersten 2 Zeilen und 3 Streifen für die letzte Zeile.

Zusätzlich zu dem Bild B enthält eine Instanz noch eine natürliche Zahl k, die angibt, wie viele verschiedene Streifen wir maximal färben dürfen. Ziel ist es, möglichst viele Felder richtig zu färben. Dabei macht es keinen Unterschied, ob wir ein Feld nicht oder falsch färben. Sind in dem obigen Beispiel nur $k = 3$ Streifen erlaubt, so ist es optimal, die erste Zeile komplett rot und die anderen Zeilen komplett grün zu färben. In diesem Fall haben wir drei Felder fälschlicherweise grün gefärbt, aber alle anderen richtig.

Geben Sie einen möglichst effizienten Algorithmus an, der mithilfe von dynamischer Programmierung für eine beliebige Instanz berechnet, welche Streifen in welcher Farbe gefärbt werden sollten, um möglichst viele Felder korrekt zu färben.