Prof. Dr. Norbert Blum
Maria Nikolaidou
Claus Rick

Institut für Informatik, Abt. IV

Vorlesung Informatik III
WS 1998/99
13. Übung

26. 1. 1999

Aufgabe 1:

Ein großes Wirtschaftsmagazin will seinen Lesern eine Analyse der Börsenentwicklung der letzten fünf Jahre präsentieren. Dazu soll unter anderem für die Kurse der wichtigsten Aktien nachträglich ein bestes Paar von Einkaufstag und Verkaufstag bestimmt werden, so daß der in der Zwischenzeit erzielte Gewinn zu einem eingesetzten Betrag maximiert wird (die Länge des Zeitraums muß nicht berücksichtigt werden, sondern nur die prozentuale Steigerung). Für jede Aktie liegen der Kurs zum Beginn des Untersuchungszeitraumes und die Folgen der absoluten Kursveränderungen im Vergleich zum Vortag vor, also etwa `127.5: -0.5 2 -1 1 3.5 -13 7 -2 -6 -9 -21 -17 -5 0.5 4 -7 -12 2.5 -3 2`. Bei diesem Beispiel wäre der beste Einkaufstag der 14. Börsentag gewesen und der beste Verkaufstag der 16. Börsentag. Der so realisierte Gewinn hätte 6.7669 % vom eingesetzten Kapital betragen.
Begründen Sie, warum man für diese Aufgabenstellung das Divide-and-Conquer Prinzip anwenden kann und entwickeln Sie einen entsprechenden Lösungsalgorithmus. Welche Laufzeit hat Ihr Verfahren? **(4P)**

Aufgabe 2:

Betrachten Sie das in der Vorlesung vorgestellte Verfahren zur Lösung des Handlungsreisendenproblems mittels dynamischer Programmierung.

a) Machen Sie sich für einen Graphen mit fünf Knoten klar, welche Lösungen von Teilproblemen zur Lösung eines größeren Teilproblems benötigt werden, indem Sie einen geschichteten Graphen zeichnen, in dem jeder Knoten ein Teilproblem repräsentiert und die Kanten die Abhängigkeiten spezifizieren. **(1P)**

b) In welcher Reihenfolge sollten die Teilprobleme bearbeitet werden, damit jedes Problem nur einmal gelöst werden muß? Sehen Sie eine Möglichkeit den Platzverbrauch, der zur Speicherung der Werte von Teillösungen benötigt wird, zu reduzieren? **(1P)**

c) Arbeiten Sie eine Implementierung des vorgestellten Verfahrens aus. Beachten Sie insbesondere, daß auf die schon bekannten Werte von Teillösungen nach Möglichkeit in konstanter Zeit zugreifbar sein sollte. **(3P)**

d) Wie muß das Verfahren erweitert werden, damit neben dem Wert der optimalen Lösung auch eine diesem Wert entsprechende Knotenfolge ausgegeben werden kann? Ist dann die in b) angedachte Speicherplatzreduzierung noch möglich? **(1P)**

Aufgabe 3:

Ein Dieb beraubt einen Laden und findet eine gewisse Anzahl von Artikeln. Jeder Artikel wird durch seinen Preis und sein Gewicht charakterisiert. Da die Rucksackkapazität des Diebes beschränkt ist, versucht er, den Rucksack möglichst optimal zu füllen. Formal läßt sich dieses Problem wie folgt definieren:
Gegeben: $p_1, \ldots, p_n$, $w_1, \ldots, w_n$, B mit $p_i, w_i, B \in \mathcal{N}$
Gesucht: Eine Teilmenge $S \subseteq \{1, \ldots, n\}$, so daß $\sum_{i \in S} w_i \leq B$ und $\sum_{i \in S} p_i$ maximiert wird.

a) Geben Sie eine optimale Lösung für folgendes Beispiel an: $n = 8$, $B = 110$, $P = \{11, 21, 31, 33, 43, 53, 55, 65\}$, $W = \{1, 11, 21, 23, 33, 43, 45, 55\}$ **(1P)**

b) Angenommen, es sind Bruchteile der Artikel zugelassen. Wie sieht dann die Greedy-Strategie aus, die das Problem optimal löst? Welche Laufzeit hat diese Strategie? **(1P)**

c) Die Greedy-Strategie scheitert, wenn keine Bruchteile zugelassen sind. Begründen Sie dies anhand eines Beispiels. **(1P)**

d) Entwickeln Sie einen dynamischen Programmierungsalgorithmus, der das Problem exakt löst, wenn keine Bruchteile zugelassen sind, und der in Zeit $\mathcal{O}(n \cdot B)$ läuft. **(4P)**

e) Haben Sie jetzt ein $\mathcal{NP}$-vollständiges Problem in polynomieller Zeit gelöst? **(1P)**

Aufgabe 4:

Entwickeln Sie einen Greedy-Algorithmus, der für einen gegebenen Graphen $G = (V, E)$ eine überdeckende Knotenmenge berechnet, die höchstens doppelt so groß wie eine minimale überdeckende Knotenmenge ist. (Tip: Ihre Greedy-Strategie sollte nicht einen einzelnen Knoten wählen, sondern eine Kante.) **(3P)**