Karlsruher Institut für Technologie
Institut für Theoretische Informatik

Prof. Dr. Dennis Hofheinz
Lukas Barth, Lisa Kohl

12. Übungsblatt zu Algorithmen I im SoSe 2016

https://crypto.iti.kit.edu/index.php?id=algo-sose16
{lisa.kohl,lukas.barth}@kit.edu

Aufgabe 1 (*Rucksackproblem,* $2+2$ *Punkte*)

Gegeben seien $n = 5$ Gegenstände mit Gewichten $w_1 = 40,\ w_2 = 18,\ w_3 = 20,\ w_4 = 15,\ w_5 = 10$ und jeweiligen Profiten $p_1 = 90,\ p_2 = 45,\ p_3 = 52,\ p_4 = 42,\ p_5 = 35$. Gesucht ist eine Teilmenge der Gegenstände, die insgesamt das Gewicht $M = 50$ nicht überschreitet und die Summe der zugehörigen Profite maximiert.

a) Finden Sie diese Teilmenge mit Hilfe dynamischer Programmierung. Verwenden Sie dazu wie in der Vorlesung vorgestellt eine Tabelle (Vorlesung vom 06.07., Folie 13).

b) Finden Sie diese Teilmenge mit Hilfe von Branch-and-Bound. Verwenden Sie dazu die Schreibweise aus der Vorlesung (Vorlesung vom 11.07., Folie 12). Sortieren Sie die Gegenstände zunächst gemäß der Annahme im vorgestellten Algorithmus. Benutzen Sie als obere Schranke den maximalen Wert, der erreichbar wäre, wenn auch Bruchteile von Gegenständen eingepackt werden dürften.

Aufgabe 2 (*Doktor Meta und dynamische Programmierung,* $2+2+4+2$ *Punkte und* 2 *Bonuspunkte*)

Dem ebenso beharrlichen wie größenwahnsinnigen Doktor Meta ist es endlich gelungen, er steht kurz vor Übernahme der Weltherrschaft. Dies möchte er gebührend feiern und beauftragt dazu seinem ebenso getreuen wie tollpatschigen Gehilfen Røbøt mit der Zusammenstellung einer Playlist. Damit diese seinen Ansprüchen genügt und um schlechte Übergänge zu vermeiden, hat Doktor Meta eine Liste von n Liedern zusammengestellt und darüber hinaus zu je zwei Liedern i und j eine Bewertung $c_{i,j}$ abgegeben, die anzeigt, wie gut Lied j klingt, wenn es direkt nach Lied i abgespielt wird. Die Aufgabe von Røbøt ist es nun eine Reihenfolge zu finden (wobei jedes Lied genau einmal gespielt werden soll), so dass die Summe der Bewertungen $c_{i,j}$ für hintereinander gespielte Lieder maximiert wird. Helfen Sie Røbøt, indem Sie berechnen, wie hoch diese Summe maximal sein kann. Zwei Bonuspunkte gibt es, wenn Sie zusätzlich die zugehörige Reihenfolge der Lieder zurückgeben. Der Algorithmus soll eine Laufzeit in $\mathcal{O}(n^2 2^n)$ haben.

a) Zählen Sie die Tabelle(n) auf, die Ihr Algorithmus verwenden wird und erklären Sie die Bedeutung der jeweiligen Einträge.

b) Geben Sie die Rekurrenz und den Basis-Fall an, den Ihr Algorithmus verwendet.

c) Geben Sie Ihren Algorithmus im Pseudo-Code an.

d) Analysieren Sie die Laufzeit Ihres Algorithmus. Begründen Sie ihre Antwort kurz.

Aufgabe 3 (*Kickern,* $1+2+4$ *Punkte und* $2+2$ *Bonuspunkte*)

In der Informatik-Fakultät wird begeistert gekickert, und so soll eine Fakultätsmeisterschaft ausgetragen werden. Da Informatiker von Natur aus kompetitiv sind, soll nachher möglichst für jedes Paar von Mitarbeitern der Fakultät feststehen, wer gegen wen gewonnen hat. Gleichzeitig haben die n Mitarbeiter (hierzu zählen auch die Professoren) aber mit dem Betreuen von Vorlesungen so viel zu tun, dass nicht daran zu denken ist, n^2 Spiele auszutragen. Daher spielt immer nur jeder Mitarbeiter gegen seinen unmittelbaren Vorgesetzten und umgekehrt, also z.B. jeder Professor gegen alle seine Mitarbeiter, usw.

Jeder Mitarbeiter außer dem Dekan hat genau einen Vorgesetzten. Für jedes ausgeführte Spiel werden die Beteiligten sowie die erreichte Tordifferenz notiert. So bedeutet z.B. der Eintrag

$$(Bob, Alice, 5)$$

dass Bob gegen Alice mit 5 Toren Differenz gewonnen hat, während

$$(Eve, Mallory, -3)$$

bedeutet, dass Eve gegen Mallory mit 3 Toren Differenz verloren hat.
Um auch für Informatiker, die nicht gegeneinander gespielt haben, festzustellen, wer gegen wen gewonnen hat, wird nach folgenden zwei Regeln vorgegangen:

- Hat Spieler A gegenüber Spieler B eine Tordifferenz von x, Spieler B gegenüber Spieler C eine Tordifferenz von y, und haben A and C nicht gegeneinander gespielt, so hat A gegenüber C die Tordifferenz $x + y$
- Hat Spieler A gegenüber Spieler B eine positive Tordifferenz (d.h. größer als 0), so hat A gegen B gewonnen.

Beispiel: Folgende Spiele haben stattgefunden:

$$(Bob, Alice, 5), (Eve, Mallory, -3), (Alice, Eve, 1)$$

Damit hat Bob gegenüber Mallory eine Tordifferenz von 3 und somit gegen Mallory gewonnen.
Im Folgenden dürfen Sie davon ausgehen, dass alle Namen maximal 100 Zeichen lang sind, und die maximale Tordifferenz einer Partie 10 beträgt.

a) *Warmup:* Folgende Spiele haben stattgefunden:
(Cormen, Garey, -6), (Garey, Johnson, -3), (Garey, Rivest, 1), (Floyd, Cormen, -4), (Floyd, Warshall, 2), (Cormen, Dijkstra, -2), (Bellman, Dijkstra, 4), (Bellman, Kruskal, 1), (Bellman, Jarnik, -2), (Bellman, Prim, 1), (Ford, Dijkstra, -2), (Ford, Karatsuba, 4), (Ford, Offman, 3), (Offman, Euler, -4), (Offman, Steiner, -3)
Nach der oben angegebenen Definition von *gewonnen*: Gegen wen hat Herr Garey gewonnen? Gegen wen steht es unentschieden?

b) Begründen Sie kurz: Die Definition der Tordifferenz ist konsistent, insbesondere kann folgender Fall nicht eintreten:

   - A hat gegenüber B Tordifferenz x
   - A hat gegenüber C Tordifferenz y
   - B hat gegen D gespielt und Tordifferenz x' gegenüber D
   - C hat gegen D gespielt und Tordifferenz y' gegenüber D
   - $x + x' \neq y + y'$

c) Skizzieren Sie einen Algorithmus, der für eine gegebene Person A und eine Ergebnisliste in $\mathcal{O}(n \log n)$ Zeit berechnet, gegen wen A gewonnen hat. Ihr Algorithmus hat dabei ausschließlich der Ergebnisliste als Eingabe zur Verfügung, und keine Zusatzinformationen wie z.B. die Liste der Mitarbeiter oder die Hierarchie innerhalb der Informatik-Fakultät. Geben Sie an welche Datenstrukturen Sie jeweils verwenden und beachten Sie die dafür anfallenden Laufzeiten.

d) *Bonusaufgabe für 2 Bonuspunkte:* Begründen Sie das Laufzeitverhalten Ihres Algorithmus.

e) *Bonusaufgabe für 2 Bonuspunkte:* Lässt sich das auch in erwartet $\mathcal{O}(n)$ berechnen? Oder sogar deterministisch in $\mathcal{O}(n)$? Wenn ja, skizzieren Sie jeweils, wie das geht. Wenn nein, begründen Sie kurz.

Ausgabe: Mittwoch, 13.07.2016
Abgabe: Freitag, 22.07.2016, 12:45 im Briefkasten im Untergeschoss von Gebäude 50.34

Deckblatt Übungsblatt 12
Algorithmen I

A1	**A2**	**A3**	$\sum$

Tutoriumsnummer:

Name	**Matrikelnummer**	**Unterschrift**

Mit unseren Unterschriften bestätigen wir, dass die Aufgaben von den Unterzeichnern eigenständig gelöst worden sind.

Hinweis: Das Übungsblatt darf in Gruppen von bis zu zwei Personen bearbeitet werden. Beide Personen müssen demselben Tutorium zugeteilt sein. Möchte jemand seine Abgaben-Gruppe innerhalb des Semesters wechseln, so ist dies im Voraus mit dem Tutor abzusprechen. **Bitte tragen Sie in das obere Quadrat *groß* die Nummer Ihres Tutoriums ein.** Die Lösung des Übungsblattes ist in jedem Fall mit diesem Deckblatt abzugeben.