Albert-Ludwigs-Universität
Institut für Informatik
Prof. Dr. Susanne Albers
Sylva Scholz

Freiburg i. Br., den 30. Mai 2005

Approximations- und Online-Algorithmen SS 05

Übung 4

Abgabe am 13. Juni 2005 (in der Vorlesung)

Aufgabe 1: Page Replication (1 + 5 Punkte)
Beim *Page Replication* Problem sind ein ungerichteter Graph $G = (V, E)$ und ein Knoten $s \in V$, in dem sich eine Datenbank befindet, gegeben. Eine Anfragesequenz besteht aus Anfragen an die Datenbank, die in den Knoten des Netzwerkes entstehen. Eine Anfrage im Knoten $x \in V$ wird dadurch bearbeitet, dass eine Verbindung zu einem Knoten $y \in V$ hergestellt wird, in dem sich die Datenbank befindet. Die Kosten für diese Verbindung sind $d(x, y)$, die Länge des kürzesten Weges zwischen x und y. Ein Algorithmus kann nun zusätzlich Kopien der Datenbank in den Knoten erzeugen. Beim Kopieren der Datenbank vom Knoten u, der bereits eine Kopie besitzt, zum Knoten v entstehen Kosten von $D \cdot d(u, v)$.

- Modellieren Sie das Ski-Ausleihe-Problem aus der 1. Übung als Spezialfall des Page Replication Problems.
  Hinweis: Der zugehörige Graph ist sehr klein und einfach.

- Zeigen Sie, dass der folgende Algorithmus $(2 - \frac{1}{D})$-kompetitiv für das Page Replication Problem in Bäumen ist.
  Hinweis: Nutzen Sie die Idee aus der Analyse des Ski-Ausleihe-Problems.

  Algorithmus Page Replication
  Sei G ein Baum mit Wurzel s. Für jeden Knoten $v \in V$ benutzt der Algorithmus einen Zähler $z[v]$, der anfangs auf Null gesetzt wird. Bei einer Anfrage im Knoten v werden alle Zähler auf dem Pfad von v zum nächsten Vorgänger im Baum, der eine Kopie der Datenbank besitzt, um Eins erhöht. Erreicht der Zähler in v den Wert $D - 1$, wird die Datenbank zu allen Knoten auf dem Weg vom nächsten Vorgänger mit einer Kopie zu v kopiert.

Aufgabe 2: Datenkompression (3 + 3 Punkte)

- Verschlüsseln Sie mit Hilfe der Burrows-Wheeler-Transformation das Wort *mississippi*.

- Entschlüsseln Sie mit Hilfe der Burrows-Wheeler-Transformation das Wort *rtwtretteae* mit Index 11.

Aufgabe 3: Scheduling (9 Punkte)
Es seien zwei verwandte Maschinen unterschiedlicher Geschwindigkeiten gegeben. Die schnellere von beiden benötigt für die Ausführung eines Jobs der Bearbeitungsdauer p genau p Zeiteinheiten und die langsamere $\alpha \cdot p$ Zeiteinheiten ($\alpha \geq 1$).
Wir betrachten die Verallgemeinerung des **LIST**- bzw. **Greedy-Algorithmus** aus der Vorlesung zur Online-Bestimmung des minimalen Makespan: Sei j der aktuelle Job und F bzw. S die aktuelle Last der schnellen bzw. langsamen Maschine. Der Algorithmus weist nun der schnellen Maschine den aktuellen Job zu, wenn $F + p_j \leq S + \alpha \cdot p_j$ gilt, d.h. derjenigen Maschine, auf der er zuerst beendet wird.
Zeigen Sie, dass der Algorithmus $\min\left\{\frac{2\,\alpha+1}{\alpha+1}, 1 + \frac{1}{\alpha}\right\}$-kompetitiv ist.

Hinweis: Führen Sie eine Fallunterscheidung nach $\alpha \leq \frac{1+\sqrt{5}}{2}$ und $\alpha > \frac{1+\sqrt{5}}{2}$ durch.

Aufgabe 4: Bin Packing (3 Punkte)
Beim *Bin Packing* Problem werden n Objekte i mit Gewicht $w_i \in (0, 1]$ geliefert. Diese Objekte sollen in eine minimale Anzahl von Kisten der Kapazität 1 gepackt werden. Wir betrachten den Online-Algorithmus **Any Fit**: Existiert für ein eintreffendes Objekt i eine offene Kiste, in die i passt, wird es in diese Kiste gepackt. Andernfalls wird eine neue Kiste geöffnet, in die i gepackt wird.
Zeigen Sie, dass **Any Fit** ein 2-kompetitiver Online-Algorithmus für das Bin Packing Problem ist.