

Effiziente Algorithmen

Sommersemester 2019

Prof. Dr. Martin Hofer
Daniel Schmand
Martin Ludwig, Conrad Schecker



Institut für Informatik
Algorithmen & Komplexität

Übung 7

Ausgabe: 28.05.2019
Abgabe: 04.06.2019, 10:15

Aufgabe 7.1. (4 Punkte)

Beschreibe einen ungerichteten Graphen $G = (V, E)$ mit Knotenmenge $V = \{v_1, \dots, v_{100}\}$, bei dem die Hitting Time $H(v_1, v_1)$ des Random Walks auf G maximal groß ist. Berechne die Hitting Time und argumentiere, warum dein Graph sie maximiert.

Aufgabe 7.2. (6 Punkte)

In der Verwaltung einer Universität müssen während eines Semesters verschiedene Prozesse J bearbeitet werden. Stell dir vor, es gibt insgesamt m Mitarbeiterinnen und Mitarbeiter im Prüfungsamt und jede/jeder kann jeden Prozess bearbeiten. Die Bearbeitungszeit p_j eines Prozesses $j \in J$ ist unabhängig von der bearbeitenden Person. Die Prozesse bauen nicht aufeinander auf, das heißt kein Prozess muss auf die Fertigstellung eines anderen warten. Jede Person kann nur einen Prozess gleichzeitig bearbeiten und jeder Prozess kann nur von einer Person bearbeitet werden. Einmal begonnene Prozesse können nicht unterbrochen werden. Das Problem ist, dass die Prozesse erst im Laufe des Semesters nach und nach bekannt werden und direkt beim Bekanntwerden eines Prozesses muss dieser zur besseren internen Planung sofort einer Person zugeordnet werden. Wir nennen die Zeit des Bekanntwerdens von Prozess j hier r_j . Zum Zeitpunkt t haben wir also nur Informationen über die Prozesse $\{j \in J \mid r_j \leq t\}$, alle anderen Prozesse sind noch unbekannt. Sobald alle Prozesse des aktuellen Semesters fertig bearbeitet wurden, kann das Semester abgeschlossen werden.

Wir suchen einen Algorithmus, der jeden Prozess direkt zum Zeitpunkt des Bekanntwerdens r_j einer Person zuordnet. Das Semester soll dabei möglichst früh abgeschlossen werden. Für eine Instanz I nennen wir diesen Zeitpunkt, bei dem alle Prozesse fertig bearbeitet wurden, $C(I)$.

Wir vergleichen nun den Wert $C(I)$ der Zuordnung des Algorithmus mit der bestmöglichen Abschlusszeit des Semesters in einer Zuordnung, bei der kein Prozess j vor der jeweiligen Zeit r_j bearbeitet wird. Diese nennen wir $C^*(I)$.

Beschreibe einen Algorithmus, der ohne Wissen über die Zukunft jeden Prozess j direkt zum Zeitpunkt r_j einer Person zuordnet und bei dem $C(I) \leq 2C^*(I)$ für alle möglichen Instanzen I . Beweise, dass die Lösung deines Algorithmus diese Schranke einhält.

Hinweis: Dieses Problem scheint ähnlich, aber nicht äquivalent zum Makespan-Scheduling auf identischen Maschinen zu sein.

Bitte wenden!

Aufgabe 7.3.

(3 + 3 Punkte)

Beim Maximum Independent Set-Problem wird für einen gegebenen ungerichteten Graphen $G = (V, E)$ eine möglichst große Menge $C \subseteq V$ gesucht, sodass für jede Kante $\{u, v\} \in E$ höchstens einer der Knoten u oder v in C liegt. Das heißt, wir suchen die größte Menge von Knoten, sodass keine zwei Knoten in C direkt mit einer Kante verbunden sind. In der Online-Variante gibt es mehrere Runden und der Graph wird nach und nach aufgedeckt. In Runde i wird genau ein neuer Knoten v_i zusammen mit allen Kanten $\{\{v_i, v_j\} \mid \{v_i, v_j\} \in E \text{ und } j < i\}$ zu bereits bekannten Knoten aufgedeckt. Wir nehmen für diese Aufgabe an, dass der Knotengrad jedes Knotens in V höchstens d ist.

- a) Zeige, dass jeder deterministische Onlinealgorithmus einen Wettbewerbsfaktor von mindestens d hat.
- b) Beschreibe einen deterministischen Onlinealgorithmus mit Wettbewerbsfaktor d . Beweise dass der Wettbewerbsfaktor deines Algorithmus höchstens d beträgt.

Dies ist das letzte Hausaufgabenblatt für die Veranstaltung Effiziente Algorithmen 1. Die letzte Vorlesung zu Teil 1 findet am Dienstag, den 28.5. statt.

Die Übungsblätter und weitere Informationen zur Vorlesung finden Sie unter <http://algo.cs.uni-frankfurt.de/lehre/ea/sommer19/ea19.shtml>

E-Mail: {mhoefer,schmand}@em.uni-frankfurt.de