Institut for Matematik og Datalogi
Syddansk Universitet

May 25, 2010
JFB

On-Line Algorithms – F10 – Lecture 9

Instruktorater

Hjælp dine medstuderende, dygtiggør dig selv og få penge for det. Søg I dag!

Der er ofte gode muligheder for at få et instruktorat, selvom man ikke er langt henne i studiet. Se nærmere information på

`http://www.jobs.sdu.dk/vis_stilling.php?id=5878&lang=da`

Hvis du allerede er ansat som instruktor i foråret 2010, leverer du ansøgning om tildeling af timer ind på IMADAs sekretariat (se opslaget vedr. bilag mm.)

Hvis du har spørgsmål, så henvend dig gerne på IMADA.

ANSØGNINGSFRIST: **02. juni 2010 kl. 12:00**.

Lecture, May 17, in U49C

We covered chapter 8 in the textbook quickly. We also covered sections 10.1 and 10.4 of chapter 10 in the textbook.

Lecture, May 25 and 27

We will cover up through and including section 12.2.2 of chapter 12 in the textbook.

Problems originally scheduled for May 27 are for May 25

The last problem should be ignored. We will probably have time to begin on chapter 12 on May 25.

Problems for May 27

1. Do exercise 7.3 in the textbook. See page 122 for the coupon collector's problem. Assume that there are $k + 1$ pages in all, and obtain kH_k as a result.

2. Work out Example 8.5, and apply Yao's principle correctly to Example 8.4 in the textbook, using the

   distribution given there. You will not get as good a result for Example 8.4 as for Example 8.5.

3. Consider the following on-line problem: We have one processor. Jobs arrive over time; job J_j with processing time p_j arrives at time r_j. A job can be assigned to run on the processor when it arrives or any time after that. It can also be started on the processor, stopped at some point, and restarted at some later point. No two jobs may be running at the same time. The goal is to minimize total completion time. Let C_j denote the completion time of job j. The total completion time is $\sum C_j$.

   Use Yao's principle to prove a lower bound on the competitive ratio of any randomized algorithm for this problem. Consider the following probability distribution on request sequences: At time 0, a job with processing time 1 arrives. At time $\frac{1}{3}$, two jobs with processing time 0 arrive. At time 1, all of the following jobs arrive with probability p (with probability $1-p$ none of them arrive): 10 jobs with processing time 0 and four jobs with processing time 1.

4. What is the complexity of the dynamic programming procedure used for computing the cost of an optimal offline algorithm for the k-server problem when the request sequence is of length n. For the special case of a uniform metric space a faster algorithm exists. What is its complexity?

5. Define and analyze a lazy version of DC for paging.

6. Exercise 10.1.

7. (Easy) Show that the makespan problem for identical machines is NP-hard.

8. Suppose that GREEDY is allowed n identical machines, while OPT is only allowed to use $m < n$ machines. Give a sequence showing that the ratio of GREEDY's performance to OPT's can be at least $1+\frac{m-1}{n}$ for the makespan problem. Then show that GREEDY can always achieve this ratio against such a bounded OPT.

9. Consider remark 12.1 on page 208. What is meant here? Why is there no problem if the loads can be greater than 1? (Do not try to prove the desired result for loads of at most 1.)

10. Define POST-GREEDY with release dates as the algorithm which assigns a new job (given at its release date) to the first processor which becomes free. (Jobs have processing times which may be unknown, and only one job

may be running on a processor at a time. There are m processors.) Show that POST-GREEDY is $(2 - \frac{1}{m})$-competitive.