Institut for Matematik og Datalogi
Syddansk Universitet

March 4, 2014
JFB

On-Line Algorithms – F14 – Lecture 8

Announcement

There will be no class on March 5 or March 6.

Lecture, March 3

We finished chapter 6 and began on the paper, "The relative worst order ratio applied to paging". In section 2, we considered definitions 1 and 2 and temporarily skipped the others. Next we covered the relative worst order ratio of FWF to LRU and considered look-ahead, comparing LRU to LRU(ℓ).

Lecture, March 12

We will cover section 3, the definitions for relatedness and weakly comparable in section 2 and Theorem 7 of section 5 of "The relative worst order ratio applied to paging". If there is time, we will cover Theorem 8 in section 5 of "The relative worst order ratio applied to paging". (If we do not cover this, the last problem for March 17 should be ignored.)

Lecture, March 13

We will breifly cover chapter 7 in the textbook, and we will cover chapter 8 in the textbook quickly. We will begin on sections 10.1 and 10.4 of chapter 10 in the textbook.

Problems for March 19

1. Using relative worst order analysis, compare First-Fit and Next-Fit for the classical bin packing problem (trying to minimize the number of

bins used). Next-Fit is the algorithm that only keeps one bin open at a time. If the current item fits there, Next-Fit places it there. Otherwise, it closes the bin (never considering it again) and opens a new bin.

2. 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.

3. 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.

4. 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.

5. 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?

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

7. Exercise 10.1.