Institut for Matematik og Datalogi
Syddansk Universitet

April 25, 2006
JFB

On-Line Algorithms – F06 – Lecture 12

Lecture, April 19

We covered chapter 7 quickly (no proofs) and covered chapter 8 in the textbook. We covered up through the definition of a metric space in chapter 9.

Lecture, April 26

We will cover up through section 9.4 of chapter 9 in the textbook (we will be skipping the remainder of the chapter).

Lecture, May 3

We will cover up through section 10.4 of chapter 10, and cover the algorithm in section 10.6.

Problems for April 25

1. 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.

2. 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.

3. Can Yao's principle be applied to the relative worst order ratio?

Problems for May 2

1. Do Exercise 9.1.

2. Explain the results in chapter in 9 with respect to the paging problem: the traversal algorithm, the lower bound, and the work function algorithm.

3. What problems would you run into in defining the classical and dual bin packing problems as metrical task systems? What changes can you make to the problem definitions to come closer to making it work?

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?