Homework 8
CSc 72700: Analysis of Algoritms
CUNY Graduate Center, Fall 2001

Due Wednesday, 12 December 2001

See the guidelines on the webpage for details about submitting homework. (If turning your homework in electronically, you can mail it directly to the grader at: `ivm3@columbia.edu`.)

Practice Problems

The problems in this section **are not to be submitted**. They are to help you understand the material, and some will appear on exams.

- Bottleneck TSP, 37.2-3 on p 974 (in the second edition, 35.2-3, p 1033).
- Non-crossing tours, 37.2-4 on p 974 (in the second edition, 35.2-4, p 1033).
- In the Albers notes, three algorithms for the on-line k-paging problem are suggested: LFU (Least Frequently Used), LRU (Least Recently Used), and FIFO (First In First Out). The first one is considered in a graded problem below. In the notes, the latter two are shown to k-competitive. Are they m competitive for $m < k$?
- Hiring exactly once, in the second edition (handout), 5.2-1, p 98.
- Inversions, in the second edition (handout), 5.2-5, p 99.
- Permute with all, in the second edition (handout), 5.3-3, p 105.

Graded Problems

These problems will be graded and should be submitted, following the guidelines on the webpage.

1. Bin Packing
   Problem 37-1 on p 983 (in the second edition, 35-1 on p 1049).

2. Deterministic k-paging algorithms
   Consider a machine with k fast memory locations and a large slow memory. The input is a sequence of page requests. If the page is fast memory, the request is satisfied. If it is not, a *page fault* occurs while some page is removed from main memory and the desired page is loaded into the empty spot. We measure the cost of an algorithm on an input sequence σ by

the number of page faults required to service all the page requests (see Albers notes, p 1-4 for more details).

Let **LFU** (Least Frequently Used) be the algorithm that on a page fault, evicts the page in memory that was used the least frequently.

(a) Show that for each k, there exists an instance σ such that

$$C_{LFU}(\sigma) > 2 \cdot C_{MIN}(\sigma)$$

(b) Fix the k, the number of fast memory registers. Show that for each $m > 0$, there exists an instance σ such that

$$C_{LFU}(\sigma) > m \cdot C_{MIN}(\sigma)$$

(c) Conclude that the algorithm LFU is not competitive (i.e. not m-competitive for any m).

3. The hat-check problem
In the second edition (handout), 5.2-4, p 98.