CS 161 Summer 1997

Handout #19

Problem Set 5 Due 3:05p, Tuesday, July 29, 1997

Readings

1. Handout 16.
2. Sections 17.1–17.3, 37.3 (skip the analysis section).
3. *Reminder:* the midterm will take place in Gates B03 on Thursday, July 24, 1997, from 1:15p to 3:05p; accordingly, there will be no lecture on that day. For midterm information, including material covered and mechanics, please consult the class WWW pages: look under *General information/Assignments*.

Exercises

Exercises are for you to work out, with the assistance of classmates and the course staff. Please don't hand in your solutions.

1. Exercise 17.1-2.
2. Exercise 17.2-5.
3. Exercise 17.3-3.
4. In problem set #4, you found a dynamic programming solution to the log-cutting problem. Consider now a greedy algorithm that cuts the log so that the maximum length of the resulting two pieces is always as small as possible. Does this algorithm minimize the total cost? If so, prove your claim; otherwise, give a counter-example.
5. Let $P_1, P_2, \ldots, P_n$ be n programs to be stored on a disk. Program P_i requires s_i kilobytes of storage, and the capacity of the disk is D kilobytes, where $D < \sum_{i=1}^{n} s_i$.
   (a) We want to maximize the number of programs held on the disk. Prove or give a counter-example: we can use a greedy algorithm that selects programs in order of nondecreasing s_i.
   (b) We want to use as much of the capacity of the disk as possible. Prove or give a counter-example: we can use a greedy algorithm that selects programs in order of nonincreasing s_i.

Problems

Your problem set grade depends only on the solutions you hand in for the problems below.

1. **[20 points]** Exercise 17.1-3.
2. **[20 points]** Exercise 17.2-4. Clearly, we expect a greedy algorithm.
3. **[15 points]** Exercise 17.3-2. Justify your generalization; as part of your justification, you may use lemma 21.2 (page 436 of the textbook) without proof.
4. **[45 points]** Let $P_1, P_2, \ldots, P_n$ be n programs to be stored on a tape. Program P_i requires s_i kilobytes of storage; the tape is long enough to hold all the programs. We know how often each program is used: a fraction π_i of requests concern program i (and so $\sum_{i=1}^{n} \pi_i = 1$). Information is recorded along the tape at constant density, and the speed of the tape drive is also constant. After a program is loaded, the tape is rewound to the beginning. If the programs are stored in the order $i_1, i_2, \ldots, i_n$ the average time required to load a program is therefore

$$\bar{T} = c \sum_{j=1}^{n} \left[\pi_{i_j} \sum_{k=1}^{j} s_{i_k} \right],$$

where the constant c depends on the recording density and the speed of the drive. We want to minimize $\bar{T}$ using a greedy algorithm. Prove or give a counter-example for each of the following: we can select the programs

   (a) **[10 points]** in order of nondecreasing s_i;
   (b) **[15 points]** in order of nonincreasing π_i;
   (c) **[20 points]** in order of nonincreasing π_i / s_i.