Massachusetts Institute of Technology
6.042J/18.062J: Mathematics for Computer Science
Professors Nancy Lynch and Tom Leighton

Tutorial 4
October 7, 1996

Problems for Tutorial 4

1 Task Scheduling with Finitely Many Processors

Consider a poset of W tasks where each task requires unit time to be completed. Let D be the length of the longest (critical) path. Show that if we have P processors then we can complete all the tasks in at most $D + \frac{W}{P}$ time steps. Show that this bound is within a factor of 2 of the optimal for any set of tasks.

2 Conflict Graphs

Consider a collection of tasks $T_1, ..., T_n$, where each T_i has a set of resource requirements S_i. (Each S_i is a subset of some universal set of resources R.) Each task needs exclusive access to its resources when it executes and each task requires unit time to be executed.

(a) In terms of the T_i's and the S_i's, define a conflict graph describing which tasks cannot execute at the same time.

(b) If each task is supposed to execute once, describe in terms of your conflict graph, a minimum-time schedule for all of them to execute. How long does it take for all the tasks to be executed?

(c) Now suppose that each task may have to be executed an arbitrary number of times. Specifically, there is a *scheduler* who interacts with a *client*. Initially, the client submits some subset of the tasks that it wants to have performed. The scheduler responds by scheduling some of these to execute immediately, and leaves the others *pending*. At each successive time step, the client submits an additional set of tasks; these cannot include any pending tasks, but can include tasks that have already been performed at previous time steps.

You are to design a strategy for the scheduler to ensure that all requested tasks eventually get performed. Try to minimize the worst-case waiting time for an individual task.

3 Conflict Graphs and Precedence

Consider again a set of tasks $T_1, \ldots, T_n$, where each has a set of resource requirements S_i. As before, each task needs exclusive access to its resources, and each task requires unit time, but in addition, there is a precedence relation between the tasks – some tasks are required to precede other tasks.

Develop a good method for scheduling all the tasks, each exactly once. Try to minimize the total schedule time.