Homework 4
CSE 551 Foundations of Algorithms
Spring 2020

April 24, 2020

Submission Instructions: Deadline is 11:59pm on 30^{th} April, 2020. Late submissions will be penalized, therefore please ensure that you submit (file upload is **completed**) before the deadline. Additionally, you can download the submitted file to verify if the file was uploaded correctly. Submit your answers electronically, in a **single PDF**, via Canvas. You can type up the answers or scan (or take pictures) your handwritten answers.

Furthermore, please note that the graders will grade 2 out of the 4 questions randomly. Therefore, if the grader decides to check questions 1 and 4,and you haven't answered question 4, you'll lose points for question 4. Hence, please answer all the questions.

Problem 1:

In class we discussed the maximum number of compatible activity selection problem, where each activity had a specific start and finish time. Two activities were considered *compatible* if their corresponding intervals (between start and finish times) did not overlap. The goal of the problem was to select the largest set of mutually compatible activities.

Consider a variation of that problem, where each activity, in addition to start and finish times, has a profit associated with it. The profit can only be made if the corresponding activity is selected. In this new version of the problem, we need to find the set of mutually compatible activities that *maximizes* the profit. Design an algorithm for the problem and analyze its complexity. Show all your work.

Problem 2:

A magnetic tape contains n programs of length $l_1, ..., l_n$. We know how often each program is used: a fraction p_i of requests to load a program concerns program i ($1 \leq i \leq n$). This ofcourse implies that $\sum_{i=1}^{n} p_i = 1$. Information is recorded along the tape at constant density, and the speed of the tape is also constant. Each time a program has been loaded, the tape is rewound to the beginning:

If the programs are held in the order $i_1, ..., i_n$, the average time required to load a program is

$$T = c \times \sum_{j=1}^{n} [p_{i_j} \times \sum_{k=1}^{j} l_{i_k}] \tag{1}$$

where the constant c depends on the recording density and the speed of the drive. Our objective is to minimize T.

- If the programs are stored in order of increasing values of l_i, will it minimize T? If your answer is yes, then prove it, else, provide an example where the strategy fails to minimize T.
- If the programs are stored in order of decreasing values of p_i, will it minimize T? If your answer is yes, then prove it, else, provide an example where the strategy fails to minimize T.
- If the programs are stored in order of decreasing values of p_i/l_i, will it minimize T? If your answer is yes, then prove it, else, provide an example where the strategy fails to minimize T.

Problem 3:

In class, we have covered two versions of the Activity Selection Problem. Now consider a new version. Suppose that the processing cost of processor P_i per job is $C_i, 1 \leq i \leq n$. If the set of jobs $J_1, ..., J_n$ is distributed among the set of processors $P_1, ..., P_n$ such that the number of jobs scheduled on Processor P_i is $x_i, 1 \leq i \leq n, x_i \geq 0$, then the total processing cost is given by,

$$x_1 \times C_1 + x_2 \times C_2 + ... + x_n \times C_n.$$

You can assume that $C_1 \leq C_2 \leq ... \leq C_n$

Design an algorithm to distribute the jobs into processors so that the overall processing cost is minimized. Show all your work.

Problem 4:

Professor Adam has two children who, unfortunately, dislike each other. The problem is so severe that not only do they refuse to walk to school together, but in fact each one refuses to walk on any block that the other child has stepped on that day. The children have no problem with their paths crossing at a corner. Fortunately, both the professor's house and the school are on corners, but beyond that, he is not sure if it's going to be possible to send both of his children to the same school. The professor has a map of his town. Show how to formulate the problem of determining whether both his children can go to the same school as a maximum flow problem.