BLG 336E Analysis of Algorithms II

Project 3

Wednesday 27th April, 2016

Due: 15^{th} May 2016, 23:00

Problem Definition
In this project, you are asked to implement an efficient algorithm to allocate tasks for a group of robots in an object manipulation scenario.

Object manipulation is a task where a number of robots are required to transport objects from their initial positions to a destination. However, since properties of the objects and capabilities of the robots in the environment differ, task allocation should be achieved efficiently considering the capabilities of the robots.

Suppose you have *n* robots and *m* objects to be transferred to a destination. Each robot is labeled from 1 to *n* and similarly, each object is denoted with 1 to *m*. Each object *i* must be assigned to exactly one robot for transferring. A robot can carry a single object at a time. You have the information for each object, describing by which robots it can be picked up, as a list L_i. Moreover, each robot (*j*) has a battery constraint which leads it to have a maximum number of objects (P_j) to be picked up during its lifetime.

Your task is to find a feasible assignment of objects to the robots. In this assignment, each object should be assigned to a single robot. However, a robot may be assigned to more than one object. You are given an input file named *input.txt*. It includes the number of objects and robots on the first line followed by the list L. The last line includes the list *P* which describes the maximum number of objects to be carried by a specific robot.

An output file should be generated at the end of the execution that contains the allocation for each object. If a valid assignment does not exist, an error message should be written to the file. Note that your algorithm may be tested on larger sized inputs as well.

Code (50 points)
Implement an efficient Network Flow algorithm that either finds a feasible allocation of all objects or determines that no such assignment exists.

Report (50 points)

- Formulate the problem properly as a Network Flow problem. Visualize your network by stating flow values. **(15 points)**
- Describe the method that you implemented for the task allocation problem **in detail**. **(15 points)**

- Discuss the time and space complexities of your algorithm with their proofs. **(20 points)**

Submission
You should be aware that the Ninova system clock may not be synchronized with your computer, watch, or cell phone. Do not e-mail the teaching assistant or the instructors your submission after the Ninova site submission has closed. If you have submitted to Ninova once and want to make any changes to your report, you should do it before the Ninova submission system closes. Your changes will not be accepted by e-mail. Connectivity problems to the Internet or to Ninova in the last few minutes are not valid excuses for being unable to submit. You should not risk leaving your submission to the last few minutes. After uploading to Ninova, check to make sure that your project appears there.

Policy: You may discuss the problem addressed by the project at an abstract level with your classmates, but you should not share or copy code from your classmates or from the Internet. You should submit your own, individual project. Plagiarism and any other forms of cheating will have serious consequences, including failing the course.

Submission Instructions: Please submit your homework through Ninova. Please zip and upload all your files. In the archived file, you must include your completed report file and all your program and header files.

All your code must be written in C++, and we must be able to compile and run on it on ITUs Linux Server (you can access it through SSH) using g++. You should supply one source code file that calls necessary routines for all questions (Multiple files are acceptable, as long as you state the compilation instructions in your report).

When you write your code, follow an object-oriented methodology with well-chosen variable, method, and class names and comments where necessary. Your code must compile without any errors; otherwise, you may get a grade of zero on the assignment.

If a question is not clear, please let the teaching assistant know by email (daltan@itu.edu.tr).