College of Computer & Information Science
Northeastern University
Fall 2017

CS5800
MS Algorithms
21 November 2017

Problem set 6 (Due Thursday, Dec 7, 11:59 pm)

- The assignment is due at the time and date specified. Late assignments will not be accepted.
- We encourage you to attempt and work out all of the problems on your own. You are permitted to study with friends and discuss the problems; however, *you must write up your own solutions, in your own words.*
- If you do collaborate with any of the other students on any problem, please do list all your collaborators in your submission for each problem.
- Finding solutions to homework problems on the web, or by asking students not enrolled in the class (or the class staff) is strictly prohibited. If you reference any source other than the textbook or class notes, please make sure you cite them in your submission.
- We require that all homework submissions be neat, organized, and *typeset.* You may use plain text or a word processor like Microsoft Word or LaTeX for your submissions. If you need to draw any diagrams, however, you may draw them with your hand.

1. (5 points) Optimal arrangement for a company retreat

You are organizing a big dinner party for your company retreat. To increase social interaction among the various departments in your company, you would like to set up a seating arrangement so that no two members of the same departments are at the same table. Show how to formulate the problem of finding a seating arrangement that meets this objective as a maximum flow problem.

Assume that the company has n departments and the ith department has m_i members. Also assume that t tables are available and the jth table has a seating capacity of c_j.

2. (5 points) Bottleneck edge

Let $G = (V, E)$ be a flow network with source s and sink t. We say that an edge e is a *bottleneck* in G if it belongs to *every* minimum capacity cut separating s from t. Give a polynomial-time algorithm to determine if a given edge e is a bottleneck in G.

3. (5 points) NBA draft

The teams in the NBA chooses new players each year in a process called “the draft”. Each year there are n teams $t_1, ..., t_n$ and $2n$ players. The league wants to change the rules of the draft so that each team will give a list of players that it is willing to get and some algorithm will match 2 players for each team, out of the list of players the team is willing to get.

For a team t_i let us denote by A_{t_i} the set of players that the team is willing to get. Show that a necessary and sufficient condition for it to be possible to give each team 2 players out of the list of players it is willing to get, is that:

$$\left| \bigcup_{i \in I} A_{t_i} \right| \geq 2|I|$$

for any subset $I \subseteq \{1, ..., n\}$.

4. (5 points) Finding rows and columns in a matrix

Let M be a matrix with n rows and n columns whose entries are either 1 or 0. Describe an algorithm that finds a minimal set I of rows and columns of M, such that any non-zero entry is in one of the rows or columns in I.

5. (2 + 3 = 5 points) Reconstructing a tree

You are asked to reconstruct the reporting hierarchy of a huge company Disorganized, Inc., based on information that is complete but poorly organized. The information is available in an n-element array, in which each element of the array is a pair (*emp*, *boss*) where *emp* is the name of an employee and *boss* is the supervisor of the employee, and n is the number of employees in Disorganized. You may assume that the names of all employees are distinct. For the company CEO, the *boss* entry is empty.

Your task is to compute a tree in which each node has the name of an employee *emp* and a parent field pointing to the node corresponding to the supervisor of *emp* (for the CEO, the parent field will point to NIL).

(a) Design an $O(n \log n)$ time deterministic algorithm for the problem. Justify the running time of your algorithm.

(b) Using hashing, design an expected $O(n)$ time randomized algorithm for the problem. Justify the running time of your algorithm.

6. (5 points) Prize Collecting Path

You are given a graph $G = (V, E)$ and a subset $S \subseteq V$ of vertices such that each vertex in S has a prize. Consider the problem of determining a simple path in G that visits the maximum number of vertices in S.

Formulate a decision version of the above problem and prove that it is NP-complete.