CS345/CS345A : Design and Analysis of Algorithms
Semester I, 2016-17, CSE, IIT Kanpur

Theoretical Assignment VI

Deadline : 6:00PM, 7th November

Important Guidelines:

- It is only through the assignments that one learns the most about the algorithms and data structures. You are advised to refrain from searching for a solution on the net or from a notebook or from other fellow students. Before cheating the instructor, you are cheating yourself. The onus of learning from a course lies first on you and then on the quality of teaching of the instructor. So act wisely while working on this assignment

- There are two exercises in this assignments. Both of them deal with the applications of maximum flow. The second exercise has two problems - one easy and one difficult. Submit solution for exactly one problem for the 2nd exercise. It will be better if a student submits a correct solution of an easy problem that he/she arrived on his/her own instead of a solution of the difficult problem obtained by hints and help from a friend. Do not try to be so greedy :-).

1. An atmospheric science experiment

Your friends are involved in a large scale atmospheric experiment. They need to get good measurements on a set S of n different conditions in the atmosphere (such as the ozone level at various places), and they have a set of m balloons that they plan to send up to make these measurement. Each balloon can make at most two measurements.

Unfortunately not all balloons are capable of measuring all conditions, so for each balloon $i = 1, \ldots, m$, they have a set of S_i of conditions that balloon i can measure. Finally, to make the results more reliable, they plan to take each measurement from at least k different balloons. (Note that a single balloon should not measure the same condition twice). They are having trouble figuring out which conditions to measure on which balloon.

1. (*marks = 35*)

   Give a polynomial-time algorithm that takes the input to an instance of this problem (the n conditions, the sets S_i for each of the m balloons and the parameter k) and decides whether there is a way to measure each condition by k different balloons, while each balloon only measures at most two conditions.

2. (*marks = 15*)

   There is one more constraint that has to be followed. Each of the balloons is produced by one of three different sub-contractors involved in the experiment. A requirement of the experiment is that there be no conditions for which all k measurements come from balloons produced by single sub-contractor. Explain how to modify your polynomial time algorithm for part (a) into a new algorithm that decides whether there exists a solution satisfying all the conditions from (a) plus a new constraint about the subcontractors described above.

1 2.

Attept exactly one of the following two problems.

1. Mobile phones and base stations

(*marks=30*)

This is the 3rd problem in Practice sheet 6. Provide complete description of the solution for this problem.

2. A social networking problem

(*marks=50*)

Cafebook is a very popular website on which many students of course CS345 spend their time whenever they get bored with the course. They send friendship requests to other students in whom they are interested. A student accepts request if he/she is also interested in the friendship. We can thus model the registered persons of Cafebook as an undirected graph $G = (V, E)$ as follows. V denote the set of persons registered on Cafebook. If a persons i and person j are mutual friends, then $(i, j) \in E$.

A set X of persons on Cafebook is said to form a *friend-group* intuitively if there are enough edges within X. We can define it formally as follows. Let $E(X)$ denote the set of edges with both endpoints belonging to X, that is, $E(X) = E \cap (X \times X)$. A subset X is said to form a *friend-group* if $|E(X)|$ is at least 10 times $|X|$. Unfortunately, the entire set V associated with Cafebook is not a *friend-group*. We wish to determine if there exists any *friend-group* on Cafebook. Design a polynomial time algorithm for this problem by formulating it in terms of max-flow in a directed graph.

Remark: Just imagine how would you have gone about solving this problem if you had no clue about max-flow. It is indeed amazing to see the varieties of problems that can be solved by their re-formulation as an instance of a max-flow problem....