CME 305: Discrete Mathematics and Algorithms

Instructor: Reza Zadeh (rezab@stanford.edu)

HW#4 – Due at the beginning of class Thursday 03/10/16

1. Let $G = (V, E)$ be a c-edge connected graph. In other words, assume that the size of minimum cut in G is at least c. Construct a graph $G'(V, E')$ by sampling each edge of G with probability p independently at random and reweighing each edge with weight $1/p$. Suppose $c > \log n$, and ϵ is such that $\frac{\log(n)}{c\epsilon^2} \leq 1$. Show that as long as $p \geq \frac{\log(n)}{c\epsilon^2}$, with high probability the size of every cut in G' is within $(1 \pm \epsilon)$ of the cut in the original graph G.

2. Let V be a finite set. A function $f\colon 2^V \to R$ is submodular iff for any $A, B \subseteq V$, we have
$$f(A \cap B) + f(A \cup B) \leq f(A) + f(B)$$
Now consider a graph with nodes V. For any set of vertices $S \subseteq V$ let $f(S)$ denote the number of edges $e = (u, v)$ such that $u \in S$ and $v \in V - S$. Prove that f is submodular.

3. A square integer matrix A is **unimodular** if and only if its determinant is -1 or 1. A matrix (not necessarily square) M is **totally unimodular** iff every square submatrix has determinant 1, -1, or 0, i.e. every non-singular square submatrix is unimodular.

   Show that for a linear program with totally unimodular constraint matrix M and integral right-hand side c, all extreme points must be integral.

4. We are given n jobs that each take one unit of processing time. All jobs are available at time 0, and job j has a profit of c_j and a deadline d_j. The profit for job j will only be earned if the job completes by time d_j. The problem is to find an ordering of the jobs that maximizes the total profit. First, prove that if a subset of the jobs can be completed on time, then they can also be completed on time if they are scheduled in the order of their deadlines. Now, let $E = \{1, \ldots, n\}$ and let
$$I = \{J \subseteq E : J \text{ can be completed on time } \}$$
Prove that $M = (E, I)$ is a matroid and describe how to find an optimal ordering for the jobs.

5. Given a list of personnel (n persons) and of list of k vacation periods, each period spanning several contiguous vacation days. Let D_j be the set of days included in the jth vacation period. You need to produce a schedule satisfying:

   - For a given parameter c, each tech support person should be assigned to work at most c vacation days total.
   - For each vacation period j, each person should be assigned to work at most one of the days during the period.
   - Each vacation day should be assigned a single tech support person.
   - For each person, only certain vacation periods are viable.

   Describe a polynomial time algorithm to generate an assignment or output that no assignment exists. Prove correctness.

6. Let G be a graph n nodes and an independent set of size $2n/3$. Give a polynomial time algorithm to find an independent set of size $n/3$ or greater – find a 1/2-approximation to the independent set in this graph.

7. The *directed* cut size is the number of *outgoing* edges from a cut S. The directed MAX-CUT problem asks for the cut with maximum directed cut size. Give a 1/4 approximation algorithm for this problem.

8. Online social networks carry a huge potential for online advertising. After a recent controversy, a popular social networking platform does not allow advertisers to target the users individually. However, it is allowed to run ads on user communities.

   Formally, let X be the set of all users on a social network, and $S_1, S_2, \ldots, S_m$ be subsets of X, where each S_i represents a user community. Notice that a user can belong to several communities. Suppose the advertiser can afford placing ads on at most k communities. The goal is to show the ads to as many users as possible, i.e. to find $S_{i_1}, S_{i_2}, \ldots, S_{i_k}$ such that $| \cup_{j=1}^{k} S_{i_j}|$ is maximized.

   Unfortunately, this problem is NP-hard and therefore we are interested in designing efficient approximation algorithms to solve it. Consider the following greedy approach: pick the k communities one at a time, and in each iteration pick the community that contains the largest number of users that have not been covered yet. In other words, choose the community that maximizes the current coverage. Show that this greedy approach yields at least $1 - (1 - 1/k)^k > 1 - 1/e$ fraction of the optimal solution.

   Hint: Let x_i denote the number of new elements covered by the algorithm in the i-th set that it picks. Also, let $y_i = \sum_{j=1}^{i} x_j$, and $z_i = OPT - y_i$. Show $x_{i+1} \geq z_i/k$ and prove by induction that $z_i \leq (1 - 1/k)^i OPT$.