CS 473: Algorithms, Spring 2018
HW 11 (due Wednesday, May 2^{nd} at 8pm)

This homework contains three problems. **Read the instructions for submitting homework on the course webpage**.

Collaboration Policy: For this home work, each student can work in a group with up to three members. Only one solution for each group needs to be submitted. Follow the submission instructions carefully.

For problems that ask for approximation algorithm, a full credit solution requires the following components:

- An algorithm that runs in polynomial time and retuns a valid solution (although sub-optimal).
- Proof of correctness and running time of the algorithm.
- Proof of approximation factor of the algorithm. This typically involve lower bounding OPT, and then obtaining an upper bound on the *value of the solution returned by your algorithm* as a function of lower bounds of OPT.

---

1. Provide a $1/2$-factor, polynomial time, approximation algorithm for the Acyclic Subgraph problem:

   Input. An directed graph $G = (V, E)$.

   Output. A maximum-cardinality set of edges $E' \subseteq E$ such that $G[E']$ is acyclic.

   Hint. *Arbitrarily number the vertices from 1 to n. Let E_+ be the edges going in an increasing direction, and E_- be those in a decreasing direction. Pick the biggest of E_+ and E_-.*

2. Recall as discussed in class, that one possible 2-approximation for the Vertex Cover problem involves solving the LP relaxation of the standard integer linear program, and rounding up to 1 every coordinate where the optimal value was at least $1/2$. This question asks you to extend this technique to the Set Cover problem:

   Input. A ground set $U = \{1, 2, \ldots, m\}$, and a collection of n subsets $S_1, \ldots, S_n \subseteq U$.

   Output. The minimum collection of these subsets which "covers" U, namely, a minimum-cardinality set $I \subseteq \{1, \ldots, n\}$ such that $\bigcup_{i \in I} S_i = U$.

   (a) Get a factor k, polynomial time, approximation algorithm for Set Cover, where k is the largest size of a subset, i.e., $k = \max_i |S_i|$.

   (b) Extend the Vertex Cover LP-rounding technique to get a factor f, polynomial time, approximation algorithm for Set Cover, where f is the maximum number of times some element appears in the subsets. (If $f_i := |\{j : S_j \ni i\}|$, then $f = \max_i f_i$.)

3. In the Max-SAT problem we are given a SAT formula φ and the goal is to find an assignment that satisfies the maximum number of clauses. Consider an oblivious randomized algorithm that sets each variable independently to TRUE with probability exactly 1/2.

   (a) Suppose the formula is a k-SAT formula where each clause has exactly k distinct literals. What is the expected number of clauses satisfied by a random assignment? Interestingly for 3-SAT, unless $P = NP$ the ratio provided by this simple algorithm cannot be improved!

   (b) Prove that for a general SAT formula, the expected number of clauses that are satisfied is at least $m/2$ where m is the number of clauses.

The remaining problems are for self study. Do *NOT* submit for grading.

- See Jeff's homework 11 from Spring 2016. `https://courses.engr.illinois.edu/cs473/sp2016/hw/hw11.pdf`