CS 787: Advanced Algorithms	**Homework 2**
Out: 2/26/19	**Due: 3/12/19**

Note: Homework should be submitted in pairs on Canvas. We will only accept PDF files.

1. **(Max Cut.)** In the *Max-Cut* problem, we are given an unweighted graph G with vertex set V and edge set F; Our goal is to partition V into V_1 and V_2, so that the total number of edges "cut" by the partition, $|F \cap (V_1 \times V_2)|$, is maximized. Consider the following randomized algorithm: for every vertex $v \in V$, independently flip an unbiased coin; Let V_1 be the set of all vertices for which the coin came up heads, and let V_2 be the remaining vertices.
   (a) What is the expected number of edges cut?
   (b) Determine an upper bound on the probability that fewer than $|F|/4$ edges were cut. Try to obtain as small a bound as you can.

2. **(Chernoff Bound.)** Extend the Chernoff bound discussed in class to the case of arbitrary random variables $X_i \in [0, 1]$ following the approach below.
   (a) Recall that a function f mapping reals to reals is said to be convex if and only if for every $a, b \in \Re$, $\frac{f(a)+f(b)}{2} \geq f(\frac{a+b}{2})$. Show that $f(x) = e^{tx}$ for any $t \in \Re$ is a convex function.
   (b) Jensen's inequality says that for any convex function f and any real-valued random variable X, $\mathrm{E}[f(X)] \geq f(\mathrm{E}[X])$. Use Jensen's inequality to show that if C is a random variable taking values in $[0, 1]$, and B is a $\{0, 1\}$ random variable with $\mathrm{E}[C] = \mathrm{E}[B]$, then for any convex f, $\mathrm{E}[f(C)] \leq \mathrm{E}[f(B)]$.
   (c) Use the above parts to reprove the Chernoff bound for sums of independent random variables over $[0, 1]$.

3. **(Longest Path.)** In the k-*Path* problem, we are given an unweighted graph $G = (V, E)$ with special nodes s and t and a target integer k; Our goal is to determine whether there exists a simple path from s to t in G with at least k intermediate nodes. This problem is NP-hard. (Think about why.) In this question you will develop an FPT algorithm for it using a technique called "color coding".
   (a) Consider the following variant of the k-path problem that we will call *Colorful* k-*Path*. In this variant, every vertex in the graph G is colored using one of k distinct colors. A path from s to t is called "colorful" if all of the intermediate nodes on this path have different colors. Note that colorful paths are always simple. In the colorful k-path problem, our goal is to determine whether there exists a colorful path from s to t in G with at least k intermediate nodes. Prove that this problem is FPT with parameter k.
   (b) Use your algorithm from part (a) to develop a randomized algorithm for the k-path problem that returns a path of length k, if one exists, with probability at least $1 - 1/\text{poly}(n)$. Your algorithm should run in time $O(2^{O(k)}\text{poly}(n))$ where n is the number of nodes in G.
   Hint: consider coloring nodes randomly and applying your algorithm from part (a).

4. **(Dual Fitting for Set Cover.)** Recall that in the *Set Cover* problem, we are given a ground set of elements $E = \{e_1, e_2, \cdots, e_n\}$ and a collection of subsets $\mathcal{S} = \{S_1, \cdots, S_m\}$ with $S_j \subseteq E$ and costs c_j for all $j \in [m]$. Our goal is to find the cheapest collection of subsets that covers the entire ground set E.

   In this problem, you will analyze the following greedy algorithm for set cover using a technique called "dual fitting".

   (i) Start with $T \leftarrow \emptyset$ and $F \leftarrow E$.
   (ii) While $F \neq \emptyset$ do:
   - Let $j \in [m]$ be the index for which $c_j/|S_j \cap F|$ is minimized. That is, set S_j has the smallest cost per element added.

- Add j to T and remove the elements in S_j from F.

(iii) Return T.

Now answer the following questions.

(a) In class we developed an LP relaxation for the set cover problem. Call this program PRIMAL. Write down the dual of the program PRIMAL. Call it DUAL.

(b) For this part, assume that all of the costs c_j are equal to 1. This is the unweighted version of the set cover problem. The greedy algorithm above provides an integral feasible solution to the program PRIMAL. Construct a corresponding feasible solution to the program DUAL such that the cost of the primal solution is within a factor of H_n of the cost of the dual solution. Here H_n is the nth Harmonic number:

$$H_n = \sum_{i=1}^{n} \frac{1}{i}.$$

Conclude that the greedy algorithm gives an $O(\log n)$ approximation for the unweighted set cover problem.

Hint: First try to construct a dual solution with cost exactly equal to the primal cost, but that violates dual constraints by a factor of at most H_n. Then rescale this dual solution so that it satisfies the dual constraints exactly.

(c) Extend your analysis from part (b) to the weighted version of set cover.