Assignment 4

Deadline: June 4

1. The *maximum cut problem* is defined as follows: Given an undirected graph $G = (V, E)$ along with a nonnegative weight $w_{ij} \geq 0$ for each $(i, j) \in E$. The goal is to partition the vertex set into two parts, U and $W = V - U$, so as to maximize the weight of the edges whose two endpoints are in different parts. Give a 2-approximation randomized algorithm for maximum cut problem.

2. In the *maximum k-cut problem*, we are given an undirected graph $G = (V, E)$, an nonnegative weights $w_{ij} \geq 0$ for all $(i, j) \in E$. The goal is to partition the vertex set V into k parts $V_1, \ldots, V_k$ so as to maximize the weight of all edges whose endpoints are in different parts (i.e. $\max_{(i,j)\in E: i\in V_a, j\in V_b, a\neq b} w_{ij}$).

Give a randomized $\frac{k-1}{k}$-approximation algorithm for the maximum k-cut problem.

3. Consider the *multicut problem in trees*. In this problem, we are given a tree $T = (V, E)$. k pairs of vertices s_i-t_i, and costs $c_e \geq 0$ for each edge $e \in E$. The goal is to find a minimum cost set of edges F such that for all i, s_i and t_i are in different connected components of $G' = (V, E - F)$.

Let P_i be the set of edges in the unique path in T between s_i and t_i. Then we can formulate the problem as the following integer program:

$$\begin{aligned} \text{minimize} \quad & \sum_{e\in E} c_e x_e \\ \text{subject to} \quad & \sum_{e\in P_i} x_e \geq 1, \quad 1 \leq i \leq k, \\ & x_e \in \{0, 1\}, \qquad e \in E. \end{aligned}$$

Suppose we root the tree at an arbitrary vertex r. Let $\mathrm{depth}(v)$ be the number of edges on the path from v to r. Let $\mathrm{lca}(s_i, t_i)$ be the vertex v on the path from s_i to t_i whose depth is minimum. Suppose we use the primal-dual method to solve this problem, where the dual variable we increase in each iteration corresponds to the violated constraint that maximizes $\mathrm{depth}(\mathrm{lca}(s_i, t_i))$.

Prove that this gives a 2-approximation algorithm for the multicut problem in trees.

4. In the *minimum-cost branching* problem we are given a directed graph $G = (V, A)$, a root vertex $r \in V$, and weights $w_{ij} \geq 0$ for all $(i, j) \in A$. The goal of the problem is to find a minimum-cost set of arcs $F \subseteq A$ such that for every $v \in V$, there is exactly on directed path in F from r to v. Use the primal-dual method to give an optimal algorithm for this problem.

5. Consider the following two algorithms for the knapsack problem:

- The greedy algorithm (pick the item with best value of $profit(i)/size(i)$);
- The algorithm that packs the maximum profit item. Prove that the algorithm that picks the better of these two solutions is a 2-approximation for the knapsack problem.

6. Suppose we are given a directed acyclic graph with specified source node s and sink node t, each arc e has an associated cost c_e. We are also given a length bound L. Give a fully polynomial time approximation scheme for the problem of finding a minimum-cost path from s to t, of total length at most L.

7. You are given a biased coin which comes up heads with probability p and tails with probability $1 - p$. What is the expected number of times you need to toss the coin before the first head appears (including the head itself)?

8. Suppose we have access to a source of unbiased random bits. This problem looks at constructing biased coins or dice from this source.

1. Show how to construct a biased coin, which is 1 with probability p and 0 otherwise, using $O(1)$ random bits in expectation.
2. Show how to sample from $[n]$, with probabilities $p_1, \ldots, p_n$, using $O(logn)$ random bits in expectation.
3. Show that the "in expectation" caveat is necessary: for example, one cannot sample uniformly over $\{1, 2, 3\}$ using $O(1)$ bits in the worst case.