CME 305: Discrete Mathematics and Algorithms

Instructor: Reza Zadeh (rezab@stanford.edu)

HW#3 – Due at the beginning of class Thursday 03/02/17

1. Consider a model of a nonbipartite *undirected* graph in which two particles (starting at arbitrary positions) follow a random walk i.e. with each time step both particles uniform randomly move to one of the neighbors. Prove that the expected time until they collide is $O(n^6)$. A collision is when both particles are on the same node at the same time step.

2. Let A be a $n \times n$ matrix, B a $n \times n$ matrix and C a $n \times n$ matrix. We want to design an algorithm that checks whether $AB = C$ without calculating the product AB. Provide a randomized algorithm that accomplishes this in $O(n^2)$ time with high probability.

3. Given a connected, undirected graph $G = (V, E)$ and a set of terminals $S = \{s_1, s_2, \ldots, s_k\} \subseteq V$, a multiway cut is a set of edges whose removal disconnects the terminals from each other. The multiway cut problem asks for the minimum weight such set. The problem of finding a minimum weight multiway cut is NP-hard for any fixed $k \geq 3$. Observe that the case $k = 2$ is precisely the minimum $s - t$ cut problem.

   Define an *isolating cut* for s_i to be a set of edges whose removal disconnects s_i from the rest of the terminals. Consider the following algorithm

   - For each $i = 1, \ldots, k$, compute a minimum weight isolating cut for s_i, say C_i.
   - Discard the heaviest of these cuts, and output the union of the rest, say C.

   Each computation in Step 1 can be accomplished by identifying the *terminals* in $S - \{s_i\}$ into a single node, and finding a minimum cut separating this node from s_i; this takes one max-flow computation. Clearly, removing C from the graph disconnects every pair of terminals, and so is a multiway cut.

   (a) Prove that this algorithm achieves a $2 - 2/k$ approximation.

   (b) Prove that this analysis is tight by providing an example graph where the approximation bound is exactly achieved.

4. Consider variants on the metric TSP problem in which the object is to find a simple path containing all the vertices of the graph. Three different problems arise, depending on the number (0, 1, or 2) of endpoints of the path that are specified. If zero or one endpoints are specified, obtain a 3/2 factor algorithm.

   Hint. Consider modifying Christofides algorithm for metric TSP.

5. Recall the *minimum vertex cover* problem: given a graph $G(V, E)$ find a subset $S^* \subseteq V$ with minimum cardinality such that every edge in E has at least one endpoint in S^*. Consider the following greedy algorithm. Find the highest degree vertex, add it to the vertex the S and remove it along with all incident edges. Repeat iteratively. Prove that this algorithm has an unbounded approximation factor i.e. for any c there exists an input graph G such that $|S| \geq c$ OPT.

6. A dominating set of a graph is a subset of vertices such that every node in the graph is either in the set or adjacent to a member of the set. The DOMINATING-SET problem is as follows: given a graph G and a number k, determine if G contains a dominating set of size k or less.

   (a) Show the DOMINATING-SET problem is NP-complete.

   (b) Obtain a $\ln(n)$-approximation to the DOMINATING-SET problem.

7. An oriented incidence matrix B of a directed graph $G(V, E)$ is a matrix with $n = |V|$ rows and $m = |E|$ columns with entry B_{ve} equal to 1 if edge e enters vertex v and -1 if it leaves vertex v. Let $M = BB^T$.

   (a) Prove that $rank(M) = n - w$ where w is the number of connected components of G.

   (b) Show that for any $i \in \{1, \ldots, n\}$,
   $$\det M_{ii} = \sum_N (\det N)^2,$$
   where $M_{ii} = M \backslash \{i^{th}$ row and column$\}$, and N runs over all $(n-1) \times (n-1)$ submatrices of $B \backslash \{i^{th}$ row$\}$. Note that each submatrix N corresponds to a choice of $n - 1$ edges of G.

   (c) Show that
   $$\det N = \begin{cases} \pm 1 & \text{if edges form a tree} \\ 0 & \text{otherwise} \end{cases}$$
   This implies that $t(G) = \det M_{ii}$, where $t(G)$ is the number of spanning trees of G. In this definition of a tree, we treat a directed edge as an undirected one.

   (d) Show that for the complete graph on n vertices K_n,
   $$\det M_{ii} = n^{n-2}.$$

8. Given an associative expression, we would like to evaluate it in some order which is *optimal*. Suppose we have as input n matrices $A_1, A_2, \ldots, A_n$ where for $1 \le i \le n$, A_i is a $p_{i-1} \times p_i$ matrix. Parenthesize the product $A_1 A_2 \ldots A_n$ so as to minimize the total cost. Assume that the cost of multiplying a $(p_{i-1} \times p_i)$ matrix by a $(p_i \times p_{i+1})$ matrix is $p_{i-1} \times p_i \times p_{i+1}$ flops.

   Note that the algorithm does not perform the actual multiplications. It just determines the best order in which to perform the multiplication operations and returns the corresponding cost. Come up with a dynamic programming formulation which takes as input matrices $A_1, \ldots, A_n$ and returns as output a single number describing the optimal cost of multiplying the n matrices together. Your algorithm should require at most $O(n^3)$ work, where n denotes the number of input matrices.

9. Prove that if G is connected, regular, and has an odd number of nodes, then G is Eularian.