Computer Science 360
Assignment 1

Due: September 19, 2018

1. Kleinberg and Tardos p. 109 # 8 (5 marks)

2. Kleinberg and Tardos p. 110 #10 (6 marks)

   Also show that for all $n = 3q + 1$ where q is a positive integer, there exists a graph with n vertices and two named vertices, u and v, that has $2^{\frac{n-1}{3}}$ shortest paths from u to v.

3. In a directed graph $G = (V, E)$, a vertex v is called middle if and only if for every vertex x in V either there exists a directed path from v to x, or there exists a directed path from x to v.

   (a) Given a directed acyclic graph G and a vertex u in V, provide an $O(|V| + |E|)$ time algorithm for determining whether or not vertex u is middle in G. (3 marks)

   (b) (This part is optional, it may be completed for bonus marks) Given an acyclic directed graph $G = (V, E)$ provide an $O(|V| + |E|)$ time algorithm for computing all the middle vertices of G. (4 bonus marks)

   (c) (When doing this part you may assume a solution to part (b) is available) Given a general directed graph $G = (V, E)$ provide an $O(|V| + |E|)$ time algorithm for computing all the middle vertices of G. (4 marks, for 2 marks provide a correct but slower algorithm)

4. Let $G = (V \cup U, E)$ be a bipartite graph such that each edge $e \in E$ has an associated weight $w(e)$. A matching for G is a subset $M \subseteq E$ such that no two edges in M share a common vertex. The weight of M is $w(M) = \sum_{e \in M} w(e)$.

   A greedy algorithm for bipartite matching could start with an empty matching M, and then repeatedly add the largest weight edge that does not share a vertex with an edge already included in M

   (a) Give an example edge weighted bipartite graph for which the above greedy algorithm will fail to find the maximum weight matching. (2 marks)

   (b) For bipartite graphs in which all the edge weights are distinct and each is a power of 2 (i.e. each weight is 2^i, for $0 \leq i$), prove that the above greedy algorithm always produces the maximum weight matching. (4 marks)