CS 5633 Analysis of Algorithms – Spring 12

3/29/12

9. Homework

Due **4/10/12** before class

1. **Adjacency Matrix**
   For the algorithms below, determine their runtime if the graph is given in an adjacency matrix. Justify your answers.
   (a) Depth-First Search
   (b) Breadth-First Search
   (c) Topological Sort (as described on the slides in class; not DFS-based)
   (d) Prim's MST algorithm
   (e) Kruskal's MST algorithm

2. **Cycles** (This question was on a PhD qualifying exam.)
   This problem is concerned with finding cycles in a directed graph $G = (V, E)$.
   (a) Describe an efficient algorithm that determines whether G has a cycle. Analyze the runtime of your algorithm.
   (b) Let u be a vertex in G. Describe an efficient algorithm that determines whether G has a cycle that includes u. Analyze the runtime of your algorithm.
   (c) Let u and v be two vertices in G. Describe an efficient algorithm that determines whether G has a cycle that includes both u and v. Analyze the runtime of your algorithm.
   (d) Let e be an edge in G. Describe an efficient algorithm that determines whether G has a cycle that includes e. Analyze the runtime of your algorithm.
   (e) Let e_1 and e_2 be two edges G. Describe an efficient algorithm that determines whether G has a cycle that includes both e_1 and e_2. Analyze the runtime of your algorithm.

3. **MST Variants**
   Let $G = (V, E)$ be an undirected, connected, edge-weighted graph.
   (a) Let T be an MST of G. Argue that you can rescale the weights of G by adding a positive constant to all of them or by multiplying them all by a positive constant without affecting the MST.
   (b) How would you find a *maximum* spanning tree of G? Describe the algorithm as well as its runtime, and justify why your algorithm is correct.

4. **Adding and Deleting Edges in an MST**
   (a) Given an MST for an undirected, connected, edge-weighted graph $G = (V, E)$. Now assume that a new edge e is added between two existing vertices. Describe how to find an MST of the new graph in time proportional to $|V|$.
   (b) Given an MST for an undirected, connected, edge-weighted graph $G = (V, E)$. Now assume that an edge is deleted from G and assume that this operation does not disconnect G. Describe how to find an MST of the new graph in time proportional to $|E|$. Justify the correctness of your algorithm.