15-211: Assignment 5 Theory Questions

Due: June. 17, 2011 in class

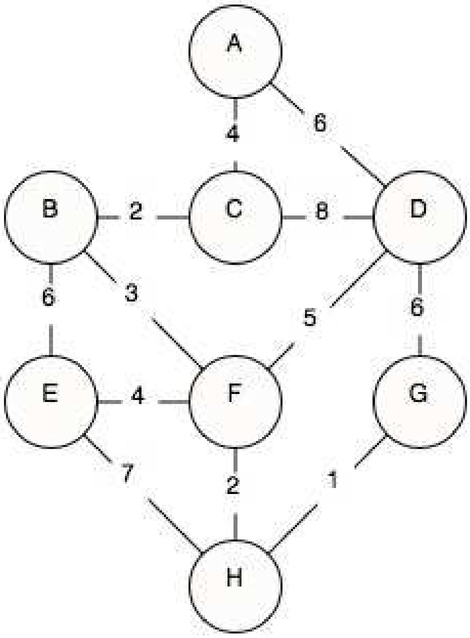
These questions count for 20% of the homework grade.

1. **DFA.**

(2) (a) Draw a DFA that accepts strings that look like 1*0*1*, where the *s are repetitions of the letter before it. It must have at least 101 but may have infinitely many more 1's after the first or last 1 and infinitely many 0s after the first 0. Example: 111011 is acceptable, 111111 is not acceptable, 1110011100 is not acceptable.

(2) 2. **Topological Sorting.** Fred Hacker has invented a new sorting algorithm. We are given n numbers a_i. Fred constructs a directed graph on n points such that there is an edge from i to j if $a_i > a_j$. He then uses Topological Sorting, a linear time algorithm, on this graph and reads off the list of sorted numbers. Fred claims his method yields a comparison based, linear time sorting algorithm. What's wrong with this approach?

3. **MST.**



(2) (a) For the graph above, list in order the edges that are added to the MST using Prim's Algorithm.

(2) (b) For the graph above, list in order the edges that are added to the MST using Kruskal's Algorithm.

(2) (c) On what kind of graphs (dense or sparse) Kruskals algorithm outperforms Prims? Give a brief analysis.

4. **Dijkstra's algorithm.**

(2) (a) Is the spanning tree formed by Dijkstras algorithm on a connected undirected graph a minimal cost spanning tree? Explain your answer.

(2) (b) Briefly explain why it is theoretically impossible for Dijkstraa algorithm to run in linear time O(V + E) regardless of what underlying data structure is used.

(2) (c) Draw a graph such that finding the shortest path with Dijkstra's algorithm would cause an infinite loop. Explain why it would cause the infinite loop.

5. **Bottleneck Problem.**

Given a directed graph, we define a bottleneck along a path as the weight of the heaviest edge. Next we define a min bottleneck path between two given vertices as a path whose bottleneck is a smallest.

(2) (a) Give an algorithm to find a min bottleneck path between two given vertices. In other words, find a path which minimizes the maximum weight edge on the path. You only need to show the modifications to Dijkstra's algorithm.

(2) (b) What is the running time of your algorithm?