CSCI 3104: Algorithms

Midterm Exam 2 (Fall 2014)

Student Name:

Student ID:

Email Address:

Honor Code Pledge: On my honor as a University of Colorado at Boulder student I have neither given nor received unauthorized assistance on this work.

_____

Student Signature

Instructions

1. This is a closed-book exam.
2. Keep one empty seat between you and your neighbor.
3. The total exam time is 45 minutes, from 11:00am to 11:45am.
4. Write down your name, student ID, email address, and sign the Honor Code Pledge.

1. Determine if the following statements are TRUE or FALSE. Briefly explain why.

   (a) Given the same graph and the same starting vertex, DFS (depth-first-search) and BFS (breadth-first-search) always visit the vertices in the same order.

   (b) No greedy algorithm can guarantee optimal solution.

   (c) Dijkstra's algorithm cannot always find the shortest path if the graph contains negative edges.

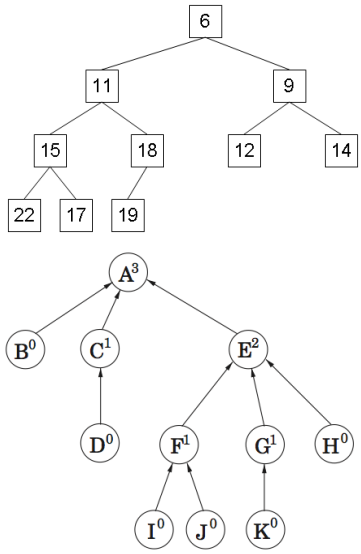
   (d) Given the same graph, Kruskal's algorithm and Prim's algorithm do not always find the same minimum spanning tree.

2. Show the key steps of the following computations:

   (a) Huffman coding of 7 symbols with the following frequencies:
   $\{A : 16, B : 3, C : 17, D : 25, E : 8, F : 19, G : 12\}$.

   (b) Given the binary heap below, show the resulting binary heap after a *delete_min*() operation.

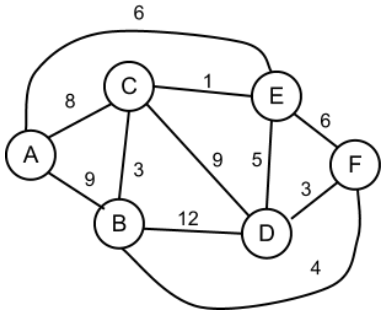
   (c) When using path compression in the $find(J)$ operation, what changes are made to the directed tree shown below?



3. Given the graph below,

   (a) When using Kruskal's algorithm to find the minimum spanning tree of the graph, which edges are selected, in which order?

   (b) When using Dijkstra's algorithm to find the shortest paths from A, in which order are the vertices examined (via *delete-min()*)? What is the shortest distance from A to each vertex?



4. **Interleaving strings**. Given three strings $A = (a_1, a_2, \ldots, a_n)$, $B = (b_1, b_2, \ldots, b_m)$, and $C = (c_1, c_2, \ldots, c_{n+m})$, determine if C can be obtained by interleaving all characters in A and B while preserving the ordering of these characters in the original A and B strings. For example, **AmSKe** can be obtained by interleaving **ASK** and **me**. Design a dynamic programming algorithm to solve this problem in $O(nm)$ time.