CSE 373: Written Assignment

Sorting and Graphs

Out: November 29th, 2017

Due: December 6th, 2017, 11:30 pm

No Late Days

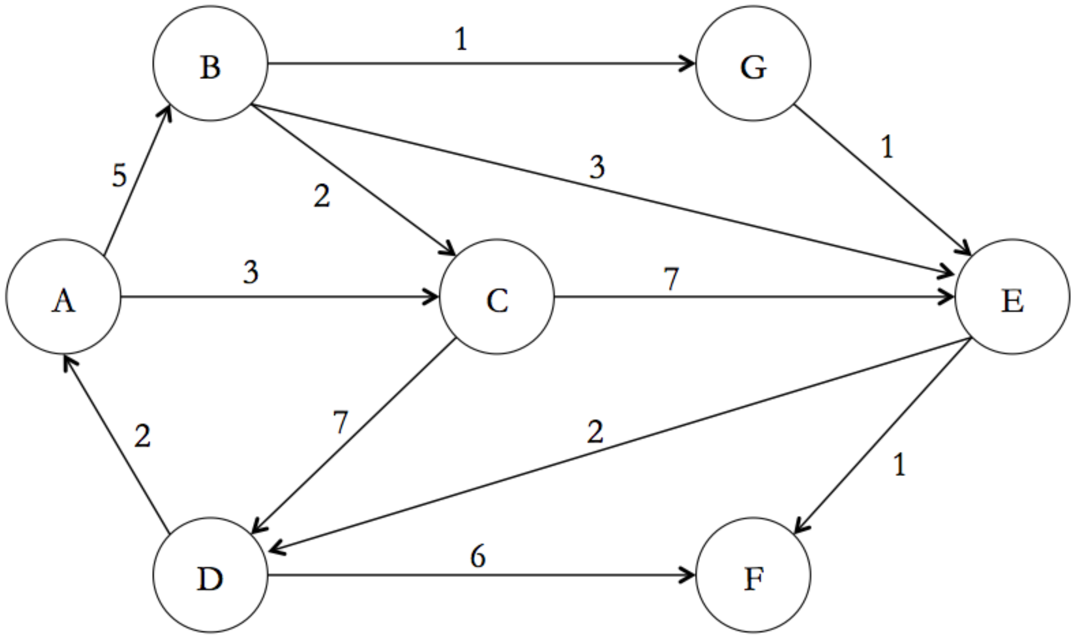
1 Quicksort

Show the Quicksort algorithm for the following data set. Use the first element as the pivot. When performing the partition, show the sub-array after each swap. If a sub-array is size 3 or less, you may just sort the elements without showing work.

$$[3, 5, 1, 7, 9, 2, 6, 4, 8]$$

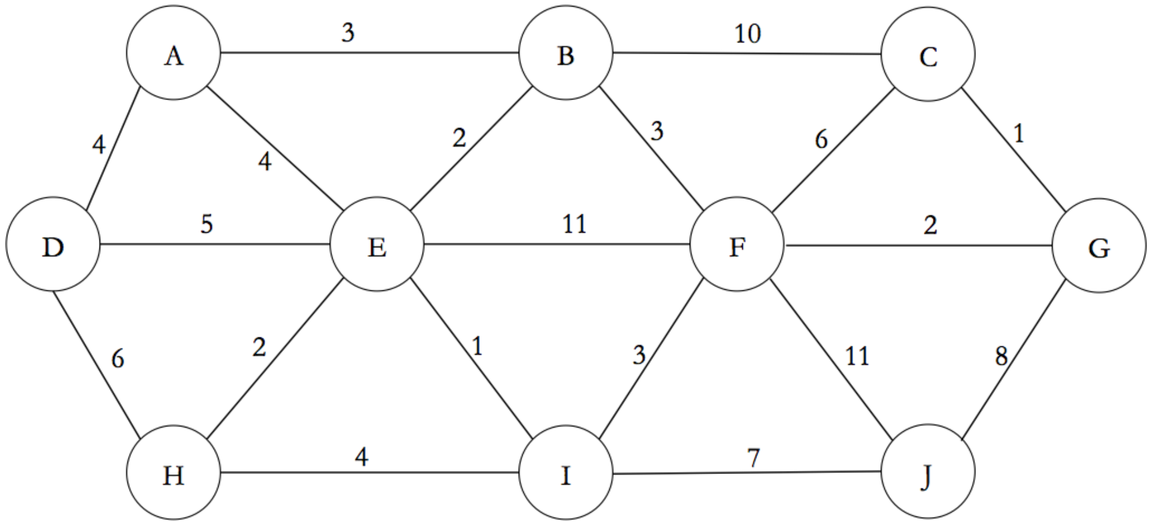
2 Dijkstra's Algorithm

Use Dijkstra's algorithm to find the costs of the shortest paths from vertex A to the other vertices. Show your work at each step and indicate the order which vertices are added to the known set. In the case of a tie, add the vertex that comes first alphabetically.



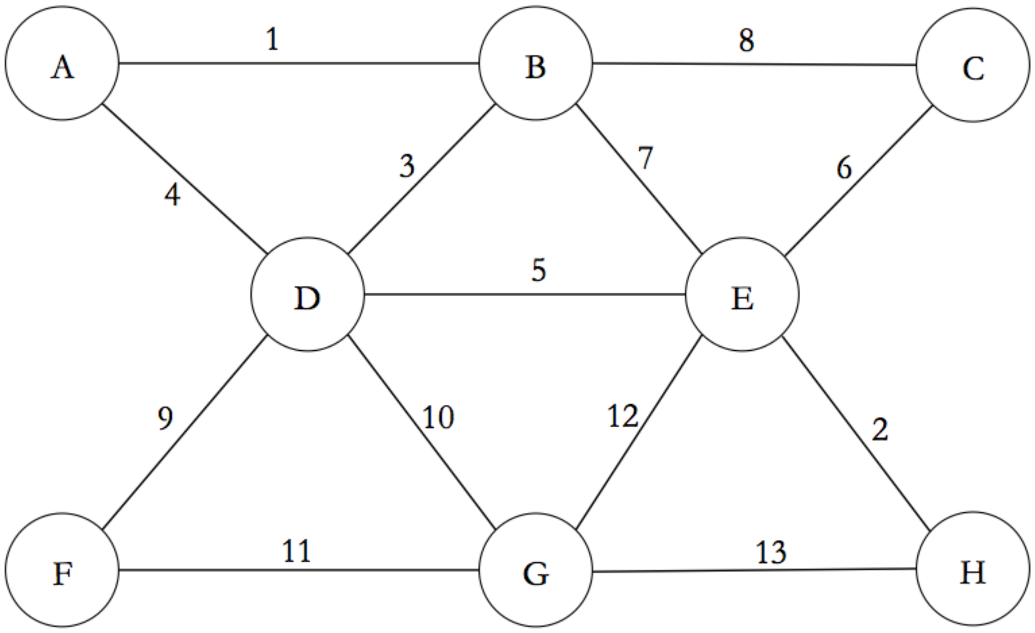
3 Prim's Algorithm

For the following graph, find the minimum spanning tree using Prim's algorithm. Start with vertex A. Show your work at each set and indicate the order in which edges are added to the MST. Break ties by choosing the vertex that comes first alphabetically.



4 Kruskal's Algorithm

For the following graph, find the minimum spanning tree using Kruskal's Algorithm. Provide *an* ordering of the edges before running the algorithm, then indicate whether that edge is added into the MST. If it is not added, give the cycle that prevents it from being added. If the algorithm terminates early, indicate when it does so.



5 Algorithm Design

Given two inputs, an array of integers n and a single integer k, design an algorithm that returns true if there are a pair of integers i, j in n such that $n[i]$ and $n[j]$ sum up to k and $i \neq j$ and false if no such pair exists.
Clearly state any reasonable assumptions that you make.
Provide a big-O worst-case analysis of your algorithm for both memory and runtime and defend your answer.

6 Extra credit

Complete a 2-4 page report on **ONE** of the following three options and submit it as a pdf to the extra credit assignment on canvas. The specs below are purposefully open-ended. The more detail you provide, the more points you are likely to receive

1. **Splay Trees:** Research and present the analysis of Splay Trees as an implementation of the Dictionary ADT. Important to include here is an amortized analysis.
2. **Skip Lists:** Research and present the analysis of the Skip List as an implementation of the Dictionary ADT. Important to include here is a probabilistic analysis.
3. **Dynamic Programming (most difficult):** Complex algorithms can take advantage of dynamic programming in order to prevent the repeated calculation of similar subproblems.

   Research and present an algorithm which solves the **coin change problem.** In addition to your algorithm, provide analysis of runtime, memory usage and correctness.