ICS 311: Algorithms Spring 2025

Problem Set 11

Kyle Berney *Due: Friday, April 4, 2025 at 4pm*

You may discuss the problems with your classmates, however **you must write up the solutions on your own** and **list the names** of every person with whom you discussed each problem.

Start **every** problem on a separate sheet of paper, with the exception of Problems 1 (Peer credit assignment) – Problem 1 can be on the same page as any other problem. Any problem that starts on the same sheet of paper as some other problem will receive 0 points!

1 Peer Credit Assignment (1 point extra credit for replying)

Please list the names of the other members of your peer group for this week and the number of extra credit points you think they deserve for their participation in group work.

- You have a total of 60 points to allocate across all of your peers.
- You can distribute the points equally, give them all to one person, or do something in between.
- You need not allocate all the points available to you.
- ***You cannot allocate any points to yourself!*** Points allocated to yourself will not be recorded.

2 Race Against the Lava Again (30 pts)

Consider again the problem about helping the geologist collect equipment near Kilauea from the previous homework (take a look at the previous homework if you need to recall the problem definition).

Design a bottom-up dynamic programming algorithm that determines the maximum possible value of the equipment that can be collected while ensuring the geologist successfully escapes from location s to t without harm. Write down the pseudocode and analyze its running time. ***Argue why your choice of the order in which you fill in the values is the correct one.*** *Hint: How can you ensure that when evaluating a location, all other reachable locations have already been processed.*

3 Building Low Cost Bridges (30 pts)

Suppose you are in Canada's Thousand Islands National Park, and in one particular lake there are n small islands that park officials want to connect with floating bridges so that people can experience going between islands without a canoe. The cost of constructing a bridge is proportional to its length. Assume the distance between every pair of islands is given to you as a two dimensional matrix (an example of such a table for $n = 8$ islands is shown below).

	A	B	C	D	E	F	G	H
A	-	240	210	340	280	200	345	120
B	240	-	265	175	215	180	185	155
C	210	265	-	260	115	350	435	195
D	340	175	260	-	160	330	295	230
E	280	215	115	160	-	360	400	170
F	200	180	350	330	360	-	175	205
G	345	185	435	295	400	175	-	305
H	120	155	195	230	170	205	305	-

Design an algorithm for determining which bridges they should build to connect the islands at minimal cost. Write down the pseudocode and explain why your algorithm correctly computes the set of bridges of minimal cost. ***No points will be given without the explanation of correctness.***

Analyze the runnning time of your algorithm. ***To receive full credit, your algorithm must be as efficient as possible.***

4 MST on Graphs with Equal-Weight Edges (20 pts)

Suppose all edge weights in a graph G are equal. What's the most efficient algorithm you can design to compute a minimum spanning tree of G? Write down the pseudocode, analyze its running time and prove that it correctly computes a minimum spanning tree of G. ***No points will be given without a proof of correctness.***

5 Divide and Conquer MST (20 pts)

Consider the following divide-and-conquer algorithm for computing minimum spanning trees:

Given a graph $G = (V, E)$, partition the set V of vertices into two sets V_1 and V_2 such that $|V_1|$ and $|V_2|$ differ by at most 1. Let E_1 be the set of edges that are incident only on vertices of V_1, and let E_2 be the set of edges that are incident only on vertices of V_2. Recursively solve a minimum-spanning-tree problem on each of the two subgraphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. Finally, select the minimum-weight edge in E that crosses the cut (V_1, V_2) and use this edge to unite the resulting two minimum spanning trees into a single spanning tree.

Prove that this algorithm correctly computes a minimum spanning tree of G, or provide an example for which the algorithm fails.

6 Kruskal's Algorithm on Restricted Range of Integer Weights (OPTIONAL - 0 pts)

How fast can you make Kruskal's algorithm run, if all edge weights are restricted as follows?

(a) All edge weights are integers in the range from 1 to $|V|$.

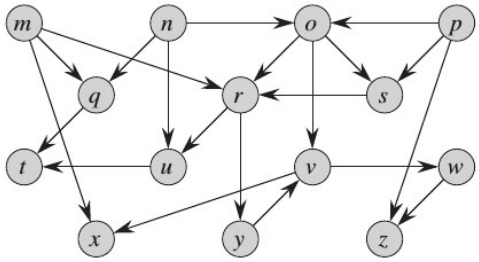
(b) All edge weights are integers in the range from 1 to c, for some constant c.

Describe what modifications you would have to implement in Kruskal's algorithm to achieve your stated runtimes.

7 Counting Simple Paths in a DAG (OPTIONAL - 0 pts)

In this problem you will design an algorithm that takes as input a directed acyclic graph $G = (V, E)$ and two vertices s and t, and returns the number of simple paths from s to t in G.

For example, the directed acyclic graph below contains exactly ***four*** simple paths from vertex p to vertex v: *pov*, *poryv*, *posryv*, and *psryv*. Notice: your algorithm needs only to count the simple paths, not list them.



(a) Design a bottom-up dynamic programming algorithm that counts the number of simple paths from s to t. Write the pseudocode and analyze its running time. ***Argue why your choice of the order in which you fill in the values is the correct one.*** *(Hint: Think about the order on which the computation of the paths from each vertex depends.)*