last name **first names** **id#**

cmput 204 assignment 4 due start of class, 2015 nov 23

Write your answers on a copy of this document. Hand in 4 separate pages, not stapled.

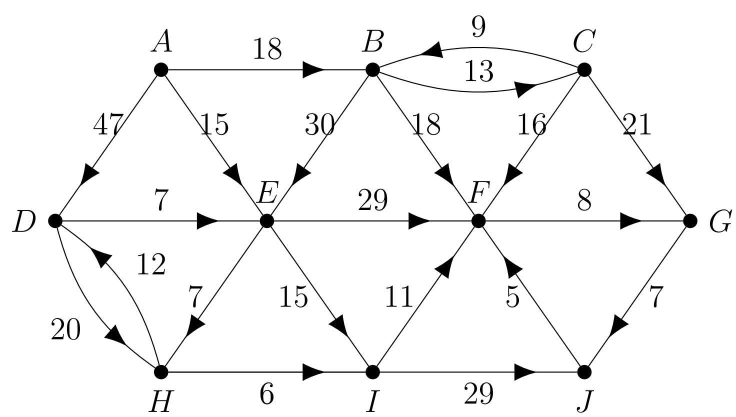
1. (a) **If you leave this question blank, your assignment will not be marked and its weight will be transferred to the final exam.** Acknowledge **all** sources, including all references and all people with whom you discussed any part of any question (for each discussion, list the relevant questions):

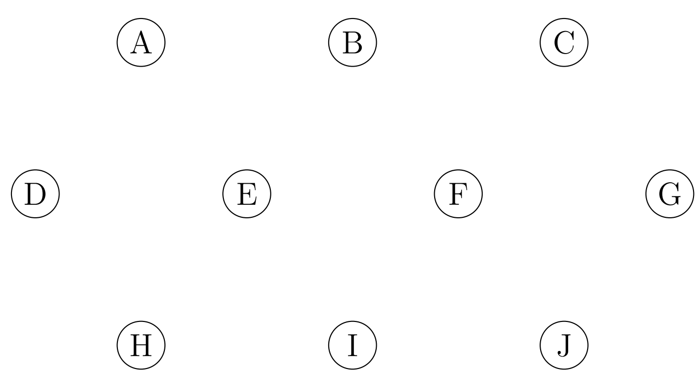
   (b) For each j, find the length of a longest increasing subsequence of L that ends with $L[j]$.

```
j        0  1  2  3  4  5  6  7  8
L       [8  3  2  7  4  9  1 11  5]
length  [1                        ]
```

   (c) Trace Dijsktra's algorithm on the digraph below, starting from A. Give the final parent and distance entries for each node. Also, on the bottom nodes, draw the final tree, and label each node in the order its distance is finalized. So, label A 0, and labe the next node whose distance is finalized 1.

```
node        A   B   C   D   E   F   G   H   I   J
parent      A
dist        0
```



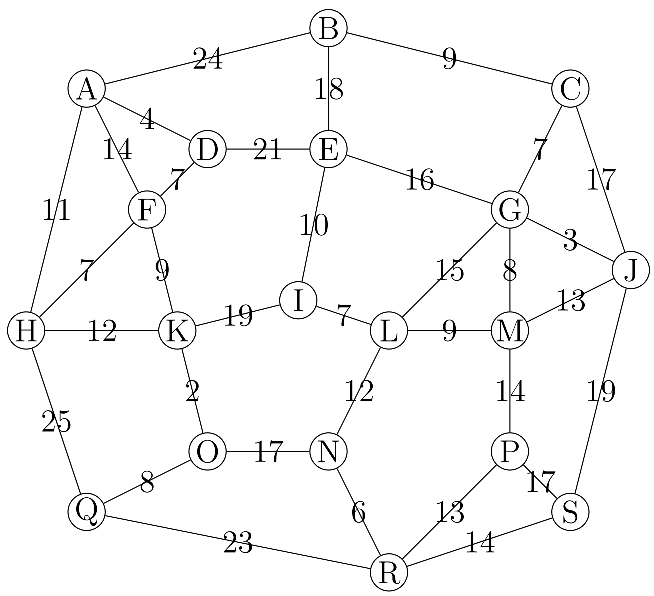


last name **first names** **id#**

2. (a) Complete the edit distance cost array for the following strings.

	-	c	a	l	b	n	o	g
-	0	1	2	3				
c	1							
l	2							
u	3							
b								
i								
n								
g								

(b) List the MST edges picked by Kruskal's algorithm in the order they are picked. If at any point there is a choice of edges to select, mention this.



(c) Repeat the previous question for Prim's algorithm, starting from I.

(d) How many MSTs does this weighted graph have? Justify briefly.

last name **first names** **id#**

3. For a graph with n nodes, let $(e_1, \ldots, e_{n-1})$ be a sequence of edges — in the order picked — from an execution of Prim's algorithm. **Claim.** For each $j \in \{1, \ldots, n-1\}$, $S_j = \{e_1, \ldots, e_j\}$ is the subset of the edges of an MST.

Proof of claim.

By induction on j. If $j = 0$ the claim holds: S_j is the empty set, which is a subset of all sets, so the subset of any MST.

Assume that the claim holds for j up to a fixed integer t. We want to show that the claim holds for $j = t+1$. So, $S_t = \{e_1, \ldots, e_t\}$ is a subset of an MST Z, since the claim holds for $j = t$.

(a) First assume e_{t+1} is in Z. Then we are done: **explain why.**

(b) Next assume e_{t+1} is not in Z. So S_t is a subset of Z, but $S_t \cup \{e_{t+1}\}$ is not. Let $Z^+ = Z + \{e_{t+1}\}$. Z^+ has a cycle C: **explain why.**

(c) Also, some edge c of $C - e_{t+1}$ has one end in S_t and one end in the rest of the input graph: **explain why.**

(d) Among all such edges c, let c^* be an edge with min cost. By Prim's algorithm, e_{t+1} is a min cost edge with one end in S_t and one end in the rest of the input graph, so $w(e_{t+1}) \leq w(c^*)$. But $Z' = Z + e_{t+1} - c^*$ is a spanning tree: **explain why.**

(e) Z is a min spanning tree, so $w(Z) \leq w(Z')$. So $w(c^*) \leq w(e_{t+1})$: **explain why.**

(f) So Z' is also a min spanning tree: **explain why.**

And Z' contains $S_t \cup \{e_{t+1}\}$, so we are done.

last name **first names** **id#**

4. (a) Matrices A, B, C, D have respective dimensions 5×2, 2×6, 6×1, 1×4. What is the minimum number of scalar multiplications needed to compute $A * B * C * D$? Give the parenthesization of matrices that yields this number. Show your work.

(b) Consider the 0-1 knapsack problem for items 1,2,3,4,5, values `[7, 6, 10, 6, 9]`, weights `[5, 8, 10, 9, 6]`, and capacity 23. Below is part of the array K (rows 10 to 20) from the webnotes dynamic programming algorithm. E.g. `K[16][4]=17` tells us that the best value knapsack with items a subset of {1,2,3,4} and total weight at most 16 has value 17. Fill in rows 20 to 23.

```
     *   1   2   3   4   5
10 [0,  7,  7, 10, 10, 10]
11 [0,  7,  7, 10, 10, 16]
12 [0,  7,  7, 10, 10, 16]
13 [0,  7, 13, 13, 13, 16]
14 [0,  7, 13, 13, 13, 16]
15 [0,  7, 13, 17, 17, 17]
16 [0,  7, 13, 17, 17, 19]
17 [0,  7, 13, 17, 17, 19]
18 [0,  7, 13, 17, 17, 19]
19 [0,  7, 13, 17, 17, 22]
20 [                     ]
21 [                     ]
22 [                     ]
23 [                     ]
```

Explain in detail how you computed `K[22][4]`.