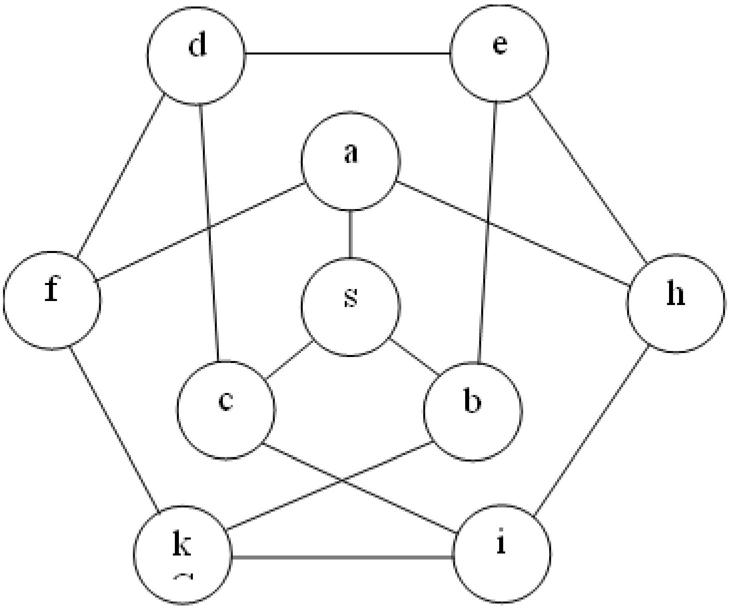
Design and Analysis of Algorithms
Fall 2021

Assignment 2

Submission: Nov. 10

Question 1 (DFS, DFS Tree) (20pt)
Given the following undirected graph G = (V, E), answer (a) to (f).



a) Represent the graph using an adjacency matrix. (2pt)
b) Represent the graph using an adjacency list. (2pt)
c) Show the tree produced by depth-first search, using vertex s as the source vertex. If multiple neighbors are available, visit them using alphabetical order. (4pt)
d) According to the DSF Tree generated in (c), writing their discovery time, finish time and draw its Time-Stamp Structure (Hint: similar to P17 in slide L6) (4pt)
e) Show the back edges with dashed lines. (3pt)
f) Find out all the articulation points and biconnected components. Based on the algorithm in P27 in slide L6, show the change process of Low[v] for each node (Hint: similar to P28 in slide L6). (5pt)

Question 2 (DFS, Backtracking) (15pt)
Given a non-empty 2D array grid of 0's and 1's, an island is a group of 1's (representing land) connected 4-directionally (horizontal or vertical.) You may assume all four edges of the grid are surrounded by water. Find the maximum area of an island in the given 2D array. (If there is no island, the maximum area is 0.) Please see the following example:
input:

[[0,0,1,0,0,0,0,1,0,0,0,0,0],
[0,0,0,0,0,0,0,1,1,1,0,0,0],
[0,1,1,0,1,0,0,0,0,0,0,0,0],
[0,1,0,0,1,1,0,0,1,0,1,0,0],
[0,1,0,0,1,1,0,0,1,1,1,0,0],
[0,0,0,0,0,0,0,0,0,0,1,0,0],
[0,0,0,0,0,0,0,1,1,1,0,0,0],
[0,0,0,0,0,0,0,1,1,0,0,0,0]]

output : 6 (marked in red).

Explanation: Note the answer is not 11, because the island must be connected 4-directionally.

a) Describe your main idea to solve this problem (2pt).
b) Show the pseudocode (C/Java code is ok) and annotation for your algorithm (10pt).
c) Analyze the running time (3pt).

Question 3 (DFS, Backtracking) (15pt)

Given a string of numbers and operators, return all possible results from computing all the different possible ways to group numbers and operators. The valid operators are +, - and *. Please see the following example:

Input: "2*3-4*5"

Output: [-34, -14, -10, -10, 10]

Note: **You may return the answer in any order.**

Explanation:

(2*(3-(4*5))) = -34

((2*3)-(4*5)) = -14

((2*(3-4))*5) = -10

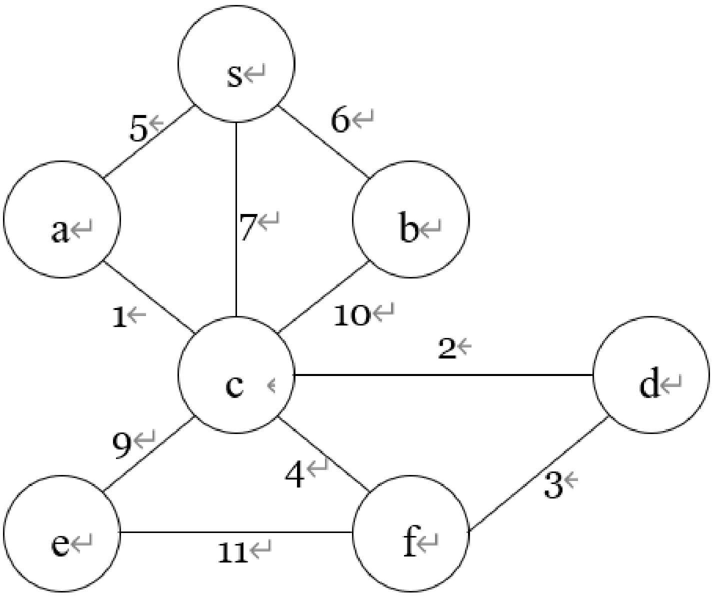
(2*((3-4)*5)) = -10

(((2*3)-4)*5) = 10

a) Describe your main idea to solve this problem (2pt).
b) Show the pseudocode (C/Java code is ok) and annotation for your algorithm (10pt).
c) Analyze the running time (3pt).

Question 4 (MST) (20pt)

Given the following weighted undirected graph G = (V, E), answer (a) to (c).

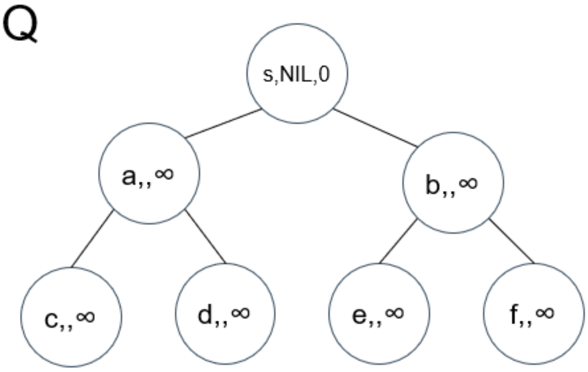


a) Consider running Prim algorithm to generate its minimum spanning tree. Show different steps the minimum spanning tree produced using node s as the root. In particular, after each step, you need to indicate key[v], pred[v], color[v] for each vertex v and the content of Q (Please refer to P32 of L7 for the definition of key, pred, color and Q) (9 pt)

Step 0, which is the initialization step, has already been done for you.

V	*s*	*a*	*b*	*c*	*d*	*e*	*f*
key[v]	0	∞	∞	∞	∞	∞	∞
Pred[v]	NIL						
Color[v]	W	W	W	W	W	W	W

Q



b) Show its minimum spanning tree produced using Kruskal's algorithm. Draw a partial forest every time an edge is added into selection (i.e., you should draw 6 forests). (6 pt)

c) Does (b) and (c) produce the same MST? If every edge in a graph has a unique weight (as in our example), does the graph have a unique MST? If your answer is yes, prove it. Otherwise, give a counter-example (i.e., a graph with unique weights having at least two different MST's). (5 pt)

Question 5 (MST Property) (20pt)

a) Show that a graph has a unique minimum spanning tree if, for every cut of the graph, there is a unique light edge crossing the cut. (6pt) Show that the converse is not true by giving a counterexample. (4pt)

b) Does the following algorithm find the minimum spanning tree T for graph G with weight w? (2pt) Please prove your conclusion. (8pt)

```
T ← ∅
for each edge e ∈ E, taken in arbitrary order
    do T ← T ∪ {e}
        if T has a cycle c
            let e′ be the maximum-weight edge on c
            T ← T − {e′}
return T
```

Question 6 (MST Application) (10pt)

In a city there are N houses, each of which is in need of a water supply. It costs W_i dollars to build a well at house i, and it costs C_{ij} to build a pipe in between houses i and j. A house can receive water if either there is a well built there or there is some path of pipes to a house with a well. Give an algorithm to find the minimum amount of money needed to supply every house with water.