CS101 Algorithms and Data Structures
Fall 2022
Homework 7

Due date: 23:59, November 13th, 2022

1. Please write your solutions in English.
2. Submit your solutions to gradescope.com.
3. Set your FULL name to your Chinese name and your STUDENT ID correctly in Account Settings.
4. If you want to submit a handwritten version, scan it clearly. `CamScanner` is recommended.
5. When submitting, match your solutions to the problems correctly.
6. No late submission will be accepted.
7. Violations to any of the above may result in zero points.

1. (?? points) Multiple Choices

Each question has **one or more** correct answer(s). Select all the correct answer(s). For each question, you will get 0 points if you select one or more wrong answers, but you will get 1 point if you select a non-empty subset of the correct answers.

Write your answers in the following table.

(a)	(b)	(c)	(d)	(e)	(f)
B	CD	ABC	D	AB	C

(a) (2') An undirected connected graph is a tree if and only if the graph

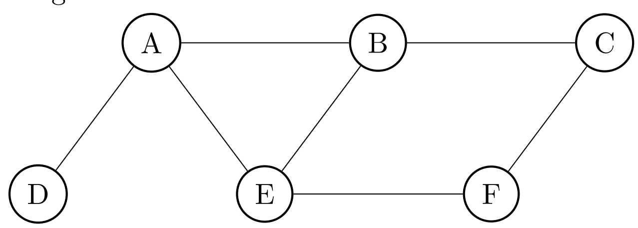
A. is a simple graph.

B. is cycle-free.

C. is a planar.

D. is bipartite.

(b) (2') If we use breadth first alogorithm to traverse the following graph, which are the possible order of visiting the nodes?



A. DABCFE

B. BAECFD

C. CBFAED

D. ADEBFC

(c) (2') Which of the following statements are true for graph traversal?

A. Given two vertices in a graph s and t, we can use both BFS and DFS to determine whether there exist a path from s to t.

B. Both DFS and BFS require $\Omega(V)$ storage for their operation.

C. Assuming we use queue to implement BFS. Let $d(v)$ be the minimum number of edges between v and the start vertex. For any two vertices u, v in the queue, $|d(u) - d(v)| \leq 2$.

D. A DFS of a directed graph always traverse through all the nodes.

(d) (2') Which of the following statements are true for graph traversal?

A. A directed graph with n nodes and $2n$ edges is strongly connected.

B. Graph with odd number of vertices cannot be a bipartite graph.

C. If a graph with n vertices has $n-1$ edges, it must be a tree.

D. Undirected graph $G = (V, E)$ is stored in an adjacency matrix A. The degree of V_i is $\sum j = 1^{|V|} A[i][j]$.

(e) (2') Consider a tree generated by disjoint set union with union-by-rank (height) strategy of height 5. Select the possible number nodes in the tree.

A. 114514

B. 32

C. 5

D. 31

(f) (2') Which of the following statements concerning the complexity of union-find data structure strategies are correct? Union-by-rank: merge the tree with smaller height into a taller one. Union-by-size: merge the tree with smaller size (number of nodes) into a bigger one.

A. When considering the asymptotic growth of the worst case running time, union-by-height is better than union-by-size.

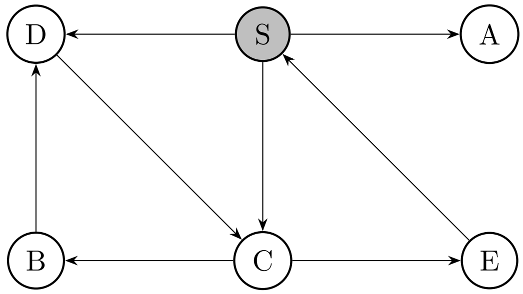
B. For a tree with n nodes generated by disjoint set union with union-by-rank, the height of the tree is $\Omega(\log n)$.

C. For a tree with n nodes generated by disjoint set union with union-by-size, the height of the tree is $O(\log n)$.

D. The worst-case running time complexity of a "find" operation in a disjoint set is so small that we can treat it as a constant.

2. (?? points) Graph traversal

Consider this directed graph starting with s.



(a) (3') Give the adjacency list for the graph. You should write the node in alphabetical order. (If no item, leave it blank).

$$\begin{aligned}
adj(S) &= [\underline{\quad \textbf{A,C,D} \quad}],\\
adj(A) &= [\underline{\qquad\qquad}],\\
adj(B) &= [\underline{\quad \textbf{D} \quad}],\\
adj(C) &= [\underline{\quad \textbf{B,E} \quad}],\\
adj(D) &= [\underline{\quad \textbf{C} \quad}],\\
adj(E) &= [\underline{\quad \textbf{S} \quad}],
\end{aligned}$$

(b) (3') Give the visited node order using the above adjacency list for Breadth First Search.

Solution: S,A,C,D,B,E

(c) (3') Give the visited node order using the above adjacency list for Depth First Search.

Solution: S,A,C,B,D,E

3. (?? points) DSU on hand

I mean, hands on DSU, perhaps.

Consider performing a series of merge opertions on a disjoint set structure employing union-by-height strategy. Draw the resulting tree structure.
When merging two sets, break tie by merging the tree whose root label is small into the other tree.

op 1. initialize: $\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{6\}, \{7\}, \{8\}$

op 2. merge 1,8

op 3. merge 2,7

op 4. merge 3,6

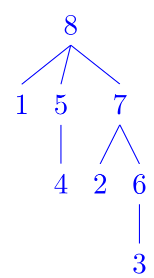
op 5. merge 4,5

op 6. merge 1,4

op 7. merge 2,3

op 8. merge 5,3

Solution:



4. (?? points) The highest DSU I've ever seen

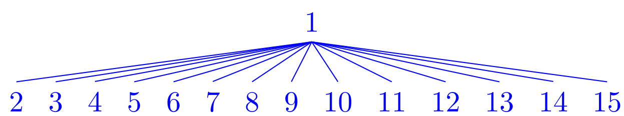
In the following tasks, you can label the nodes by whatever mark you want. We only care about the tree structure.

(a) (2') Plot a union tree of 15 nodes with minimum height. The tree was generated by disjoint-set-union with union-by-height.

(b) (2') Plot a union tree of 16 nodes with maximum height. The tree was generated by disjoint-set-union with union-by-height.

Solution:

for (a)



for (b)