

CS101 Algorithms and Data Structures  
Fall 2020  
Homework 8

---

Due date: 23:59, November 16, 2020

1. Please write your solutions in English.
2. Submit your solutions to [gradescope.com](https://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 according problem numbers correctly.
6. No late submission will be accepted.
7. Violations to any of the above may result in zero grade.

**1: (4×1') Single Choice**

The following questions are single choice questions, each question has **only one** correct answer. Select the correct answer.

*Note: You should write those answers in the box below.*

Question 1	Question 2	Question 3	Question 4
<b>D</b>	<b>D</b>	<b>A</b>	<b>C</b>

**Question 1.** Undirected Graph  $G = (V, E)$  is stored in an adjacency matrix  $A$ . We let  $A[i][j]=1$  if and only if there is an edge between  $V_i$  and  $V_j$  in  $G$ , otherwise  $A[i][j]=0$ . We want to know whether there is a path **with length  $m$**  between  $V_i$  and  $V_j$  by visiting  $P[i][j]$ ; if  $P[i][j]=0$ , then there is no such path. What should  $P$  be?

(A)  $A$

(B)  $mA$

(C)  $A^{m-1}$

(D)  $A^m$

**Question 2.** Which of following statements is **NOT** true?

(A) In a directed simple graph, the maximal number of edges is  $|V|(|V| - 1)$ .

(B) If Graph  $G = (V, E)$  is a tree, then  $|V| = |E| + 1$ .

(C) In BFS, let  $d(v)$  be the minimum number of edges between a vertex  $v$  and the start vertex. For any two vertices  $u, v$  in the fringe,  $|d(u) - d(v)|$  is always less than 2.

(D) If undirected Graph  $G = (V, E)$  is stored in an adjacency matrix  $A$ , then the degree of  $V_i$  is  $\sum_{i=1}^{|V|} A[i][i]$ .

**Question 3.** A union tree, used in a disjoint set with only union-by-size optimization, has height 6. The number of nodes contained in that tree **can not** be -----.

(A) 32

(B) 63

(C) 64

(D) 80

**Question 4.** Which of the following statements is true?

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

(B) A directed graph with  $n$  vertices has at least  $2n$  edges to ensure the whole graph is strongly connected.

(C) Both the time complexity of DFS and that of BFS are  $\Theta(|V| + |E|)$ .

(D) Every edge is visited exactly once in one iteration of DFS on a connected, undirected graph.

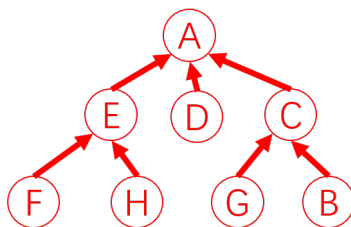
**2: (2×2') Disjoint set practice**

Given the following set of operations on a disjoint set, show the final disjoint set tree for each of the following optimization strategies.

$set\_union(A, D)$ ,  $set\_union(C, B)$ ,  $set\_union(F, E)$ ,  $set\_union(G, C)$ ,  
 $set\_union(D, G)$ ,  $find(A)$ ,  $set\_union(H, E)$ ,  $set\_union(E, G)$ ,  $find(E)$

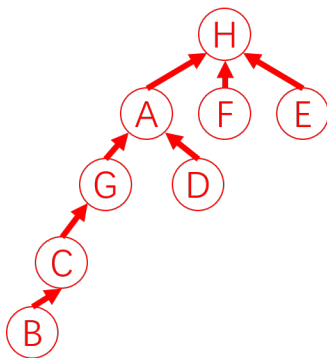
(1)(2') Only with union-by-size optimization. (When two trees have the same height, the set specified first in the union will be the root of the merged set.)

As below.



(2)(2') Only with path compression. (The set specified first in the union will always be the root of the merged set.)

As below.

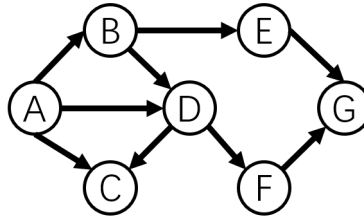


---

**3: (2×2') Graph Traversal**

---

Run depth first search (DFS) and breadth first search (BFS) on the graph below, starting from node A. List the order in which each node is first visited. Whenever there is a choice of which node to visit next, follow the alphabetical order.



(1)(2') DFS.

A B D C F G E

(2)(2') BFS.

A B C D E F G

---

**4: (4') DFS went wrong!**

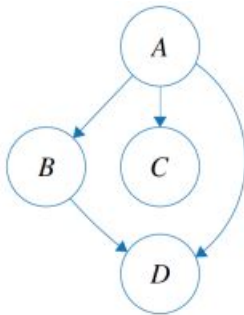
---

The following algorithm which runs DFS on a directed graph, but it contains an fatal error.

```
Create a stack.  
Choose the initial vertex and mark it as visited.  
Put the initial vertex onto the stack.  
while the stack is not empty:  
    Pop a vertex V from the top of the stack.  
    for each neighbor of V:  
        if that neighbor is not marked as visited:  
            Mark that neighbor as visited.  
            Push that neighbor onto the stack.
```

Please give a graph as an counterexample and briefly explain why this algorithm is wrong.

This is incorrect because it actually marks all neighbors of the initial node. If this happens:



If A follows the alphabetical order, then the result should be A, B, D, C.

But in this case it will be A, B, C, D.

---

**5: (2×2') Graph Algorithms**

---

For each of the following two problems, design an algorithm to solve that problem efficiently.

Give a worst case runtime bound in  $\Theta$  notation in terms of  $V$  and  $E$ .

(1) Determine whether there is a path between from node  $s$  to node  $t$  in a directed graph.

Run DFS or BFS starting from  $s$ . After that, check whether  $t$  is visited. Time:  $\Theta(|V| + |E|)$

(2) Determine whether or not a cycle exists in a directed graph.

Run DFS. If a node encountered is in the call stack (can be maintained as a disjoint set), there is a cycle.

Runtime:  $\Theta(|V| + |E|)$