CS 101 Fall 2022 - Quiz 6-A

14/11/2022 - 20 Minutes **Name:** **ID number:**

---

1. (8 points) True or False

For each statement, choose T if the statement is correct, otherwise, choose F.

Note: You should write down your answers in the box below.

(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
T	T	F	F	F	F	T	T

(a) (1') The time complexity of graph traversal on a connected graph is $\Theta(E)$ where E is the number of edges.

(b) (1') A directed graph with n vertices has n edges may be strongly connected.

(c) (1') In a simple undirected graph with $3n$ vertices and 3 connected components, the maximum number of edges in the graph is $\frac{3n(n-1)}{2}$

(d) (1') It's convenient to use a queue to implement DFS.

(e) (1') Union-Find/Disjoint-Set-Union can be used to determine reachability(whether a directed path from a given vertex to another vertex exists) on a directed graph.

(f) (1') Union-Find/Disjoint-Set-Union cannot be used to solve connected component size problem (couting vertices in a connected component) on a undirected graph.

(g) (1') A union tree generated by a union-by-rank DSU of n nodes has $O(\log n)$ height.

(h) (1') A union tree generated by a union-by-rank DSU of height h has $\Omega(2^h)$ nodes.

The **Union-by-rank** strategy: When merging two sets A, B whose root is u and v respectively, if $\text{height}(u) < \text{height}(v)$, make u a child of v. That is: merge the tree with lower rank into the other one.

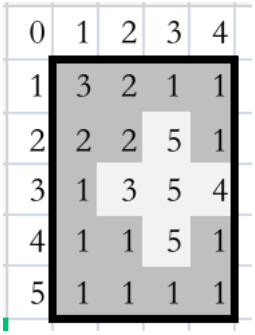
2. (5 points) Flooding

Background(you can skip this): Affected by the southeast wind, the coastal areas of our country often flood. The government wants a new system to help them protect against floods. You are asked to design a search algorithm for the system to find safe locations in a flooded area.

Goal: The geographical information for the entire area is stored in a $(n \times m)$-sized matrix, where each cell holds the altitude(height) of the location. The flood will enter the area from the (1,1) position, and the water will only flow to the cell no higher than its own altitude. Design an algorithm to find areas that won't be flooded. Briefly explain your algorithm with **natural language**.

(hint: you should try to build a graph first.)

Example: In the following graph, your algorithm should output: $(2,3),(3,2),(3,3),(3,4),(4,3)$.



Solution: Possible solution 1:

Build a directed graph first: Each cell can be viewed as a vertex in the graph. Connect each adjacent cell throught which water can flow.

Run BFS/DFS on the directed graph started from (1,1), mark any visited node.

Output all the nodes that haven't been marked.

Possible solution 2:

Build a undirected graph first: Each cell can be viewed as a vertex. Each cell are connected to its naighbor.

Modify the existing DFS/BFS: When searching, only add to the stack/queue the nodes that are connected and have less or equal height.

Run the modified DFS/BFS and output all the nodes that haven't been marked.

Possible wrong answer:

Output all nodes that are higher than all its neighbor.