

TF Questions

A directed graph has a topological ordering if and only if it contains no cycle.

Both BFS and DFS can be used to find shortest path from one node to another node on graphs that are unweighted.

If a directed graph has a topological ordering, then this topological ordering is unique.

Kruskal's, Prim's and Reverse Delete algorithms are all examples of greedy algorithms

In an unweighted strongly connected (directed) graph, the shortest distance from A to B is always the same as that from B to A.

In a weighted undirected graph, the shortest distance from A to B is always the same as that from B to A.

In an unweighted directed graph, the shortest distance from A to B is always the same as that from B to A.

In the stable matching problem with n men and n women, if a man and a woman are each other's last preference, then they will never be matched with each other in any stable matching.

We can find the k -th largest element in a binary max heap in $\Omega(1)$ time.

A strongly connected (directed) graph cannot be a DAG

If the heaviest weight edge e in an undirected connected graph G is unique, then e cannot belong to any minimum spanning tree of G .

The height of a complete binary tree with n nodes is $O(\log n)$

Given a graph G , if there is no negative cost cycles in G , then Dijkstra's algorithm will work correctly on G

MC questions: (correct answers are in red)

Which one of the following statements is False about binary heaps.

- a) We should use a min-heap instead of a max-heap in Dijkstra's algorithm to find single source shortest path.
- b) Inserting a new element into a heap has a worst-case runtime of $O(\log n)$
- c) Checking the max element in a max-heap has a worst-case runtime of $O(1)$
- d) Finding an arbitrary element in a heap has a worst-case runtime of $O(\log n)$
- e) Popping the maximum element from a max-heap has a worst-case runtime of $O(\log n)$

Running time of insertion sort on every input is:

- a) $\Theta(n^2)$
- b) $\Omega(n^2)$
- c) $O(n^2)$
- d) None of the above

Which of the following are true about the Gale-Shapley algorithm (with men proposing)?

- a) Men end up with their worst valid partners
- b) Women end up with their worst valid partners
- c) Men end up with their best valid partners
- d) Women end up with their best valid partners

Select the pair of runtimes that have the same Big-Theta

- (a) $\log n + n \log n$
- (b) 2^n
- (c) $n + \log(n^n)$
- (d) $\log(2^n)$

Select all runtimes that are $\Omega(n)$ and $O(n^3)$

- (a) $n \log n$
- (b) n^2
- (c) n^4
- (d) $\log n$

Which of the following bounds on the worst-case runtime complexity of Dijkstra's algorithm (using binary min-heap) is correct?

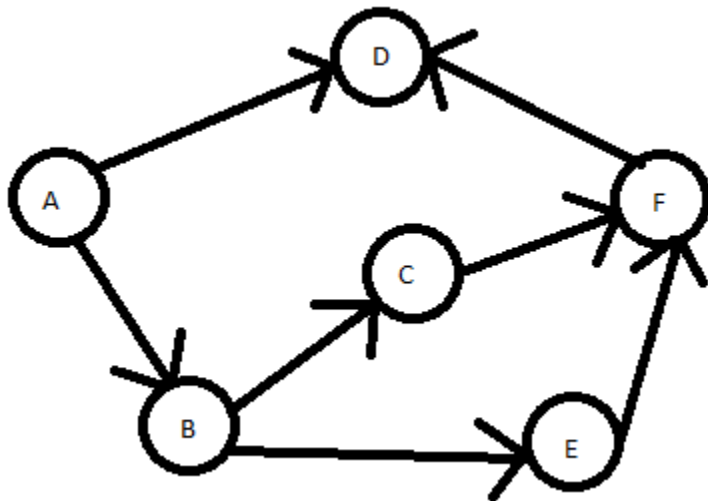
- (A) $O(V)$
- (B) $O(E)$
- (C) $O(V \log V)$
- (D) $O(E \log E)$

(E) None of the above

Suppose $[2, 5, 7, 8, 10, 9, 11]$ is a binary min-heap. What will the heap look like after running the Extract-Min operation?

- (a) $[5, 7, 8, 10, 9, 11]$
- (b) $[5, 8, 7, 11, 10, 9]$**
- (c) $[5, 7, 8, 9, 10, 11]$
- (d) $[5, 8, 7, 9, 10, 11]$

Problem 1



- a. Is it possible to get a Topological Sorted order of the graph shown above?
- b. If you answered No, justify why it is not possible for the graph.
If you answered Yes, give all possible topological sorted orders of the graph.

Problem 2A

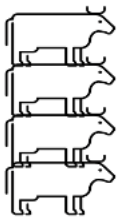
Suppose you are given an undirected weighted graph with a source node s and a destination node d . Given a particular edge e of the graph, give a polynomial time algorithm to check whether there exists a shortest path from s to d going through e .

Problem 2B

The diameter of a graph is defined as the maximum of the length of shortest paths between any pair of vertices. Design an algorithm to find the diameter of a connected undirected graph in $O(mn)$.

Problem 3

Farmer John has N cows ($1, 2, \dots, N$) who are planning to escape to join the circus. His cows generally lack creativity. The only performance they came up with is the “cow tower”. A “cow tower” is a type of stunt in which every cow (except for the bottom one) stands on another cow’s back and supports all cows above in a column. The cows are trying to find their position in the tower. Cow i ($i = 1, 2, \dots, N$) has weight W_i and strength S_i . The “risk value” of cow i failing (R_i) is equal to the total weight of all cows on its back minus S_i .



We want to design an algorithm to help cows find their positions in the tower such that we minimize the maximum “risk value” of all cows. For each of the following greedy algorithms either prove that the algorithm correctly solves this problem or provide a counter example. Hint: One of the two solutions is correct and the other is not.

- a) Sort cows in ascending order of S_i from top to bottom
- b) Sort cows in ascending order of $S_i + W_i$ from top to bottom

Problem 4

You are hosting an online auction that's open to millions of people. After your team presents a new item, you will wait for 30 minutes for people to privately send in their bid prices. In other words, during this 30 minutes, you will receive a data stream of integers like: 3100, 200, 4030, 12000, 399... To get a better sense of how people are feeling about this item, the eager analyst on your team wants you to design an efficient algorithm that can output the median bid price you have seen so far at any given time during this 30 minutes. The complexity of one query for current median has to be $O(1)$. And if you are maintaining some internal data structures, the runtime of maintaining that data structure cannot be greater than $O(\log n)$ per new bid price. Hint: See if this can be done with two heap data structures.