## **Question 1**

Consider a directed graph G=(V,E) with non-negative edge lengths and two distinct vertices sss and t of V. Let P denote a shortest path from sss to t in G. If we add 10 to the length of every edge in the graph, then: [Check all that apply.]

* P definitely remains a shortest s−t path.
* P definitely does not remain a shortest s−t path.
* If P has only one edge, then PPP definitely remains a shortest s−t path.
* P might or might not remain a shortest s−t path (depending on the graph).

## **Question 2**

What is the running time of depth-ﬁrst search, as a function of nnn and mmm, if the input graph G=(V,E) is represented by an adjacency matrix (i.e., NOT an adjacency list), where as usual n=|V| and m=|E|?

* θ(n^2)
* θ(n∗m)
* θ(n^2logm)
* θθ(n+m)

## **Question 3**

What is the asymptotic running time of the Insert and Extract-Min operations, respectively, for a heap with n objects?

* Θ(log⁡n) and Θ(1)
* Θ(logn) and Θ(logn)
* Θ(1) and Θ(logn)
* Θ(n) and Θ(1)

## **Question 4**

On adding one extra edge to a directed graph G, the number of strongly connected components...?

* ...might or might not remain the same (depending on the graph).
* ...cannot change
* ...cannot decrease by more than 1
* ...cannot decrease

## **Question 5**

Which of the following statements hold? (As usual nnn and mmm denote the number of vertices and edges, respectively, of a graph.) [Check all that apply.]

* Breadth-first search can be used to compute the connected components of an undirected graph in O(m+n) time.
* Depth-first search can be used to compute the strongly connected components of a directed graph in O(m+n) time.
* Depth-first search can be used to compute a topological ordering of a directed acyclic graph in O(m+n) time.
* Breadth-first search can be used to compute shortest paths in O(m+n) time (when every edge has unit length).

## **Question 6**

When does a directed graph have a unique topological ordering?

* None of the other options
* Whenever it has a unique cycle
* Whenever it is a complete directed graph
* Whenever it is directed acyclic

## **Question 7**

Suppose you implement the operations Insert and Extract-Min using a *sorted* array (from biggest to smallest). What is the worst-case running time of Insert and Extract-Min, respectively? (Assume that you have a large enough array to accommodate the Insertions that you face.)

* Θ(n) and Θ(n)
* Θ(n) and Θ(1)
* Θ(1) and Θ(n)
* Θ(logn) and Θ(1)

## **Question 8**

Which of the following patterns in a computer program suggests that a heap data structure could provide a significant speed-up (check all that apply)?

* Repeated minimum computations
* Repeated maximum computations
* None of the other options
* Repeated lookups

## **Question 9**

Which of the following patterns in a computer program suggests that a hash table could provide a significant speed-up (check all that apply)?

* Repeated minimum computations
* Repeated maximum computations
* None of the other options
* Repeated lookups

## **Question 10**

Which of the following statements about Dijkstra's shortest-path algorithm are true for input graphs that might have some negative edge lengths? [Check all that apply.]

* It may or may not correctly compute shortest-path distances (from a given source vertex to all other vertices), depending on the graph.
* It is guaranteed to correctly compute shortest-path distances (from a given source vertex to all other vertices).
* It is guaranteed to terminate.
* It may or may not terminate (depending on the graph).