Week 4 Quiz

TOTAL POINTS 10

1. Which of these algorithms can be used to count the number of connected components in a graph? 1 point

   - Count the number of times a breadth first traversal is started on every vertex of a graph that has not been visited by a previous breadth first traversal.
   - Count the number of times a depth first traversal is started on every vertex of a graph that has not been visited by a previous breadth first traversal.
   - All of the above
   - None of the above

2. Which elements encountered by a breadth first search can be used to detect a cycle in the graph? 1 point

   - Discovered edges that were previously unexplored by the traversal have been added to the breadth-first traversal.
   - Unexplored edges to unexplored vertices that remain so after completion of the breadth first search.
   - Previously visited vertices that have been encountered again via a previously unexplored edge.
   - Unexplored vertices that have been encountered by the traversal of a previously unexplored edge.

3. A breadth first traversal starting at vertex v1 of a graph can be used to find which ones of the following? 1 point

   - The shortest path (in terms of # of edges) between vertex v1 and any other vertex in the graph.
   - The shortest path (in terms of # of edges) between any two vertices in the graph.
   - All of the above.
   - None of the above.

4. Which traversal method has a better run time complexity to visit every vertex in a graph? 1 point

   - Breadth First Traversal
   - Depth First Traversal
   - Both have the same run time complexity.
   - Neither traversal method will necessarily visit every vertex in a graph.

5. The breadth first traversal of a connected graph returns a spanning tree for that graph that contains every vertex. If the graph has weighted edges, which of the following modifications is the simplest that produces a minimum spanning tree for the graph of weighted edges. 1 point

   - The queue is replaced by a priority queue that keeps track of the total weight encountered by the current traversal plus each of the edges that connects a vertex to the current breadth first traversal.
   - No modification is necessary because a breadth first traversal always returns a minimum spanning tree.
   - An ordinary breadth first traversal is run from each vertex (as its start vertex) and the resulting spanning tree with the least total weight is the minimum spanning tree.

- ◯ The queue is replaced by a priority queue that keeps track of the least-weight edge that connects a vertex to the current breadth first traversal.

6. True of false: a connected directed graph with no cycles is a tree. 1 point

- ◯ True
- ◯ False

7. For which situation described here can Dijkstra's algorithm sometimes fail to produce a shortest path? You would want to avoid using Dijkstra's algorithm in this situation. 1 point

- ◯ A connected graph where all of the edges have the same positive weight.
- ◯ A connected graph where some of the edge weights are zero and the rest are positive.
- ◯ A connected graph where there are multiple paths that have the same overall path cost (distance), and all of the edge weights are non-negative.
- ◯ A connected graph where some of the edge weights are negative and some have weight zero.

8. Which of the following is a true statement about Dijkstra's algorithm? Assume edge weights (if any) are non-negative. 1 point

- ◯ Dijkstra's algorithm finds the shortest unweighted path, if it exists, between a start vertex and any other vertex, but only for an undirected graph.
- ◯ Dijkstra's algorithm finds the shortest weighted path, if it exists, between a start vertex and any other vertices, but only for an undirected graph.
- ◯ Dijkstra's algorithm finds the shortest weighted path, if it exists, between a start vertex and any other vertices in a directed graph.
- ◯ Dijkstra's algorithm finds the shortest weighted path, if it exists, between all pairs of vertices in a directed connected graph.

9. Which of the following is the optimal run time complexity to find the shortest path, if it exists, from a vertex to all of the other vertices in a weighted, directed graph of n vertices and m edges. 1 point

- ◯ O(n)
- ◯ O(m + lg n)
- ◯ O(m + n)
- ◯ O(m + n lg n)

10. Suppose you are given an undirected simple graph with unweighted edges, and for a particular specification of three vertices u, v, and w, you want to find the shortest path from u to w that goes through v as a landmark. What is the most efficient method that can find this? 1 point

- ◯ Two runs of Dijkstra's algorithm, first from u and then from v.
- ◯ A single run of breadth-first search from v.
- ◯ Three runs of breadth-first search: once each from u, v, and w.
- ◯ A single run of Dijkstra's algorithm from u.

☐ I, **Jiarong Yang**, understand that submitting work that isn't my own may result in permanent failure of this course or deactivation of my Coursera account.

Learn more about Coursera's Honor Code

Save

Submit