X

(https://swayam.gov.in) (https://swayam.gov.in/nc_details/NPTEL)

regentishamitha@gmail.com

NPTEL (https://swayam.gov.in/explorer?ncCode=NPTEL) » Design and analysis of algorithms (course)

Week 4 Quiz

The due date for submitting this assignment has passed.

Due on 2021-09-22, 23:59 IST.

Score: 8/10=80%

Assignment submitted on 2021-09-22, 18:06 IST

All questions carry equal weightage. You may submit as many times as you like within the deadline. Your final submission will be graded.

1) We are given a directed graph, represented as an adjacency list. We would like to print out for each vertex `j` the list of vertices pointing into `j`. What is the most accurate upper bound on the complexity of computing these quantities in terms of `n`, the number of vertices, and `m`, the number of edges? (Note that some vertices may have no incoming and/or outgoing edges.) ***2 points***

- $O(n^2)$
- O(m)
- O(n+m)
- O(n)

No, the answer is incorrect.
Score: 0

Feedback:

In one pass of the original adjacency list, we can build an new set of lists where we associate with each vertex the set of neighbours with edges pointing into the vertex. This can be done in time O(m). If we allowed disconnected vertices, it could be that m is 0, but we still need O(n) time to set up the lists. So the correct answer is O(n+m).

Accepted Answers:

O(n+m)

Week 3 Programming Assignment

Week 4 : Weighted graphs

Week 4 Quiz

- **Quiz: Week 4 Quiz (assessment?name=125)**

Week 4 Programming Assignment

Week 5: Data Structures: Union-Find and Heaps

Week 5 : Divide and Conqure

Week 5 Quiz

Week 6: Data Structures: Search Trees

Week 6: Greedy Algorithms

Week 6 Quiz

Week 6 Programming Assignment

Week 7: Dynamic Programming

Week 7 Quiz

Week 7 Programming Assignment

2) Consider the following strategy to convert an undirected graph with negative edge weights to one that does not have negative edge weights. Let the maximum magnitude negative edge weight in the graph be -k. Then, for each edge in the graph with weight w, update the weight to w+k+1. Consider the following claim: ***2 points***

- To solve the shortest path problem in the original graph, we can run Dijkstra's algorithm on the modified graph and subtract the added weights to get the original distances.

Which of the following is ***not correct***.

- The claim is not true in general.
- The claim is not true in general for graphs with cycles.
- The claim is true for connected acyclic graphs.
- The claim is true for all graphs.

Yes, the answer is correct.
Score: 2
Feedback:
Adding a weight to each edge increases the weight of a path proportional to the number of edges, so "long" shortest paths get penalized. In a tree (connected acyclic graph), there is only one path between any pair of nodes, so the shortest path is invariant

Accepted Answers:
The claim is true for all graphs.

3) Consider the following algorithm on a connected graph with edge weights. ***2 points***

- Sort the edges as $[e_1,e_2,...,e_m]$ in decreasing order of cost.
- Start with the original graph. Consider each edge e_j. If this edge is part of a cycle in the current graph, update the graph by deleting e_j.

Which of the following is ***not true***.

- At most n-1 edges will be deleted.
- After processing all m edges, the resulting graph is connected.
- What remains at the end is a minimum cost spanning tree.
- Exactly m-n+1 edges will be deleted.

Yes, the answer is correct.
Score: 2
Feedback:
This is reverse version of Kruskal's algorithm. Any edge that is part of a cycle can be deleted without disconnecting the graph. In the end we get a minimum cost spanning tree with exactly n-1 edges.

Accepted Answers:
At most n-1 edges will be deleted.

4) Consider the following strategy to solve the single source shortest path problem with edge weights from source s. ***2 points***

1. Replace each edge with weight w by w edges of weight 1 connected by new intermediate nodes
2. Run BFS(s) on the modified graph to find the shortest path to each of the original vertices in the graph.

Which of the following statements is correct?

Week 8: Linear Programming and Network Flows

Week 8: Intractability

Week 8 Quiz

Text Transcripts

Books

Download Videos

- This strategy will solve the problem correctly and is as efficient as Dijkstra's algorithm.
- This strategy will solve the problem correctly but is not as efficient as Dijkstra's algorithm.
- This strategy will only work if the graph is acyclic.
- This strategy will not solve the problem correctly.

Yes, the answer is correct.
Score: 2
Feedback:
The strategy is sound, but the size of the graph will be proportional to the edge weights in the original graph, so the complexity will increase.
Accepted Answers:
This strategy will solve the problem correctly but is not as efficient as Dijkstra's algorithm.

5) Suppose we have a graph with negative edge weights. We take the largest magnitude negative edge weight -k and reset each edge weight w to w+k+1. Which of the following is true? ***2 points***

- Kruskal's algorithm will identify the same spanning tree on the new graph as on the old graph.
- Minimum cost spanning trees in the original graph will not correspond to minimum cost spanning trees in the new graph.
- Prim's algorithm will not work on the modified graph but will work on the original graph.
- There are more minimum cost spanning trees in the modified graph than in the original graph.

Yes, the answer is correct.
Score: 2
Feedback:
The ascending order of edge weights is unchanged if each edge weight is increased by k+1, so Kruskal's algorithm runs in exactly same way as before and produces the same tree. In general, since every spanning tree has exactly n-1 edges, the increase in weight uniformly affects all spanning trees, so the minimum cost spanning trees are unchanged. Prim's algorithm is not affected by negative weights.
Accepted Answers:
Kruskal's algorithm will identify the same spanning tree on the new graph as on the old graph.