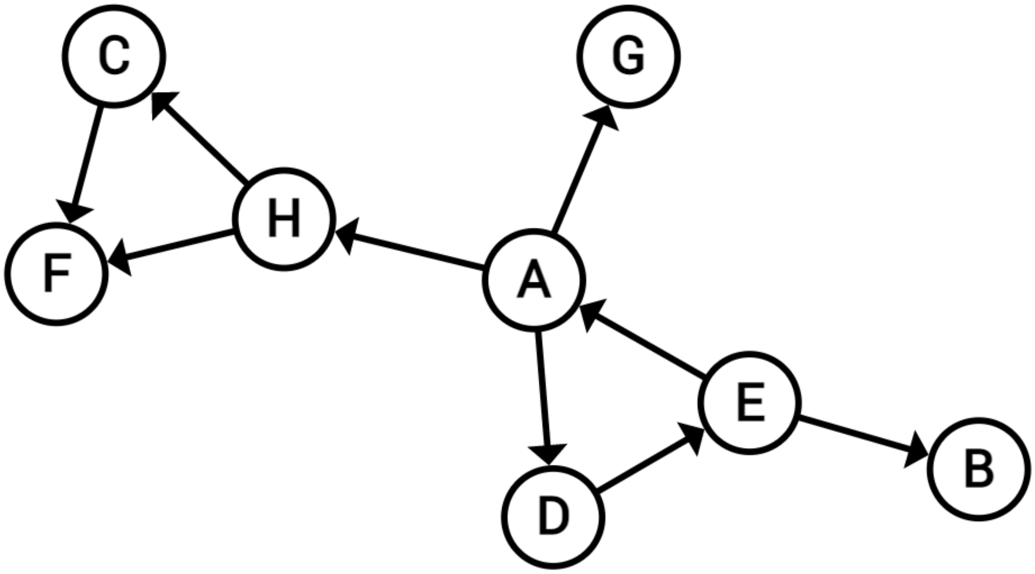
Q1 Graph Traversals

2 Points

Please look at the following graph and answer questions. If a node has more than one neighbor, visit the neighbors in alphabetical order. For example, when you iterate over the neighbors `{A, F, D, Z}` of some node, it visits A, D, F, Z in order.



Q1.1 DFS

1 Point

Starting at node A, in which order will the nodes in the graph be visited by depth-first search?

To get full points, use Capital Letters to denote nodes, and separate your letters with commas and spaces (e.g. "X, Y, Z")

Save Answer

Q1.2 BFS

1 Point

Starting at node A, in which order will the nodes in the graph be visited by breadth-first search?

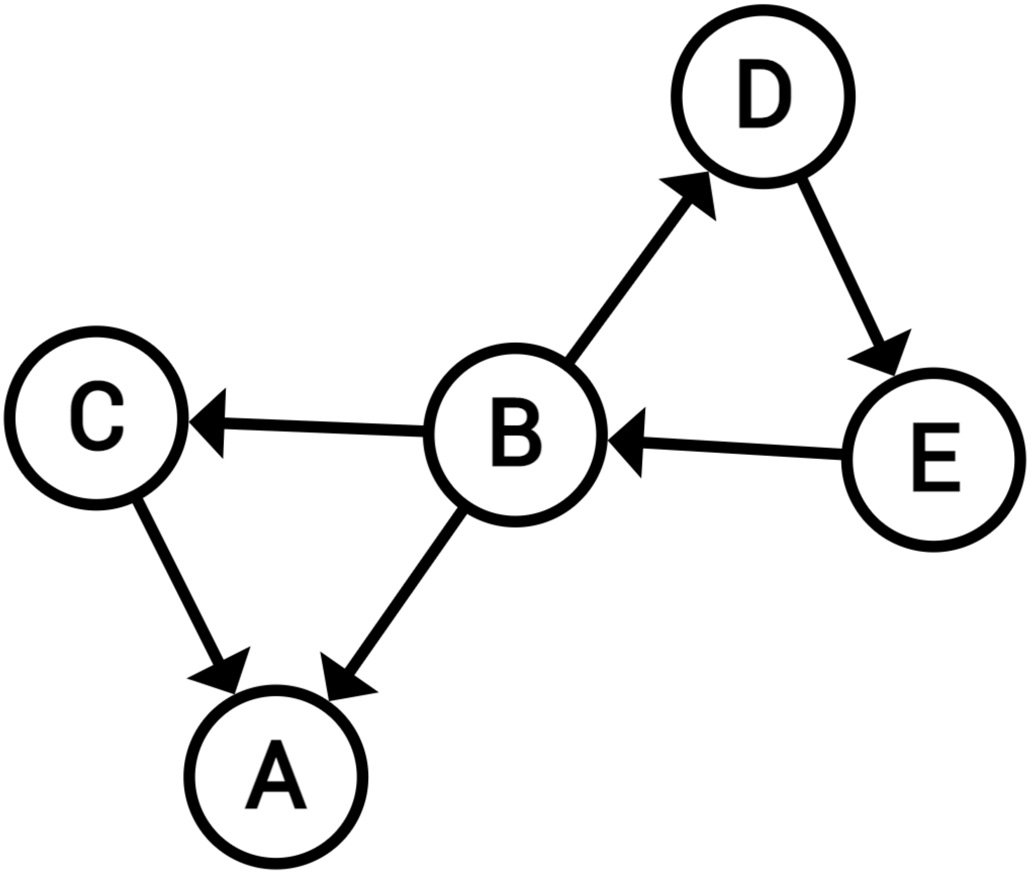
To get full points, use Capital Letters to denote nodes, and separate your letters with commas and spaces (e.g. "X, Y, Z")

Save Answer

Q2 Graph Terminology

2 Points

Which of the following features apply to the graph below?



□ Undirected

□ Directed

□ Acyclic

□ Cyclic

□ Connected

□ Strongly connected

Save Answer

Q3 Shortest Paths: T/F

5 Points

Suppose we run Dijkstra's algorithm from a start node s in a **weighted undirected graph**.

The distance to each reachable node t is given by `distTo.get(t)`.

Let u and v be two other nodes in the graph that are not the start node. For each of the following statements, select whether it is true or false.

Q3.1 T/F #1

1 Point

There can only exist a single shortest path from u to v.

True

False

Save Answer

Q3.2 T/F #2

1 Point

The shortest path from u to v must include one or more edges in the shortest paths tree (SPT) from s.

True

False

Save Answer

Q3.3 T/F #3
1 Point

If the shortest paths tree reaches u and v, then u and v are connected.

True

False

Save Answer

Q3.4 T/F #4
1 Point

The distance of a shortest path from u to v cannot be greater than `distTo.get(u) + distTo.get(v)`.

True

False

Save Answer

Q3.5 TF #5
1 Point

In a graph that contains some negatively-weighted edges, but no negative cycles, Dijkstra's algorithm will always compute the correct shortest path from the start node s to any node in the graph.

True

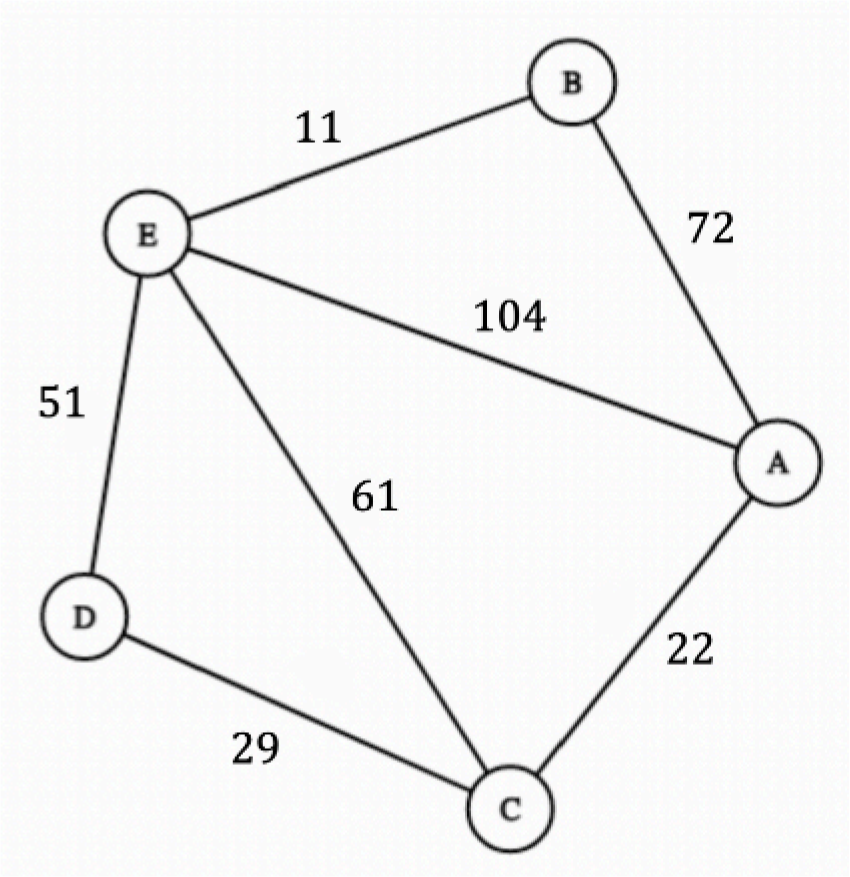
False

Save Answer

Q4 Minimum Spanning Trees

2 Points

Consider the graph below.



Q4.1 Kruskal's Algorithm

2 Points

What order are edges added to the MST when running **Kruskal's algorithm**?

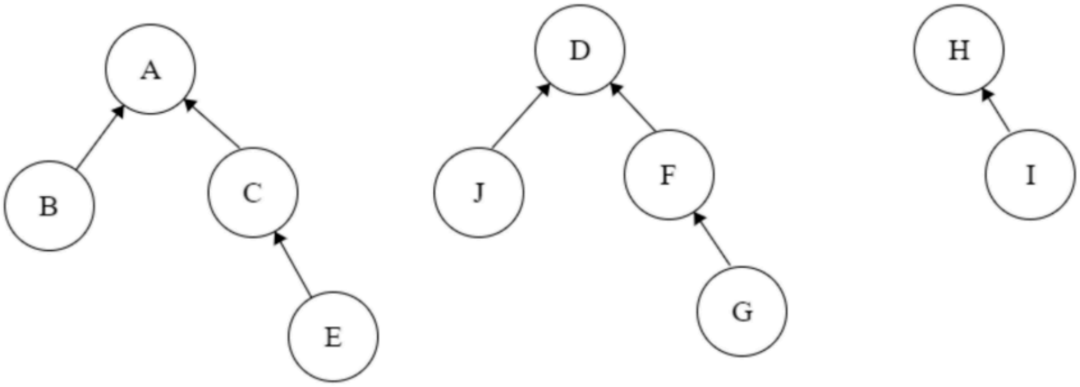
Please enter your answer as the edge weight followed by a comma separating values (e.g. `72, 61, 22, 104`). Thank you.

Save Answer

Q5 Disjoint Sets

4 Points

Consider the Array WeightedQuickUnion (with Path Compression) implementation of disjoint sets. We show the abstract view of the state as up-trees in the image below.



Consider the Array Representation of the disjoint sets above as such:

A	B	C	D	E	F	G	H	I	J

Note that each index in the array corresponds to a node in the disjoint sets (that is, the index 0 corresponds to the node A, the index 1 corresponds to the node B, and so on).

Q5.1 Array Representation
2 Points

Fill in the array representation for the disjoint sets above.

Note that you should write the index as a number corresponding to the node since arrays are indexed as such. The array picture above shows the mapping from which value goes to which index.

Enter your answer below as comma-separated values in the order as the values appear in the array above.

Save Answer

Q5.2 Calling Operations
2 Points

Now provide the array representation for the disjoint sets after calling `union(G, H)` on the disjoint sets above.

Enter your answer below as comma-separated values in the order as the values appear in the array above.

Save Answer

Q6 Graph Modelling

5 Points

Let's say you want to model a collection of Java files in a project. You know which files exists and which other Java files they depend on using. You want to determine the order that the files have to be complied in before a given file `f` can be complied and run.

What type of graph would you construct in order to create this model (i.e. directed vs. undirected, cyclic vs. acyclic)? What information do the nodes represent? What information do the edges represent? What information are you storing and how are you storing such information in your graph? How will you traverse your graph and create an **efficient** graph algorithm that returns files in the correct order such that all files can be compiled and run? Justify your answer by providing a worst-case runtime.

Briefly describe your graph model and graph algorithm in no more than 5 sentences. Thank you!

Save Answer

Save All Answers

Submit & View Submission >