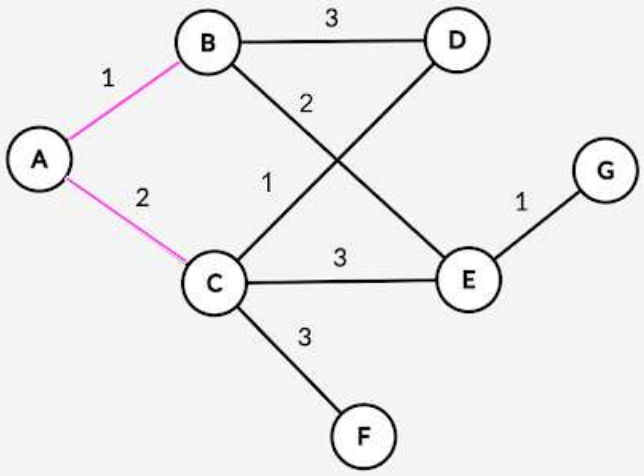
Suppose we are running Dijkstra's on the graph below, starting at source vertex A. So far, we've visited A & B. * 6 points

After we visit C, what would the new distance to each vertex be?

The pink edges represent the Shortest Paths Tree right before visiting C.

Node	distTo	edgeTo
A	0	-
B	1	A
C	2	A
D	4	B
E	3	B
F	∞	-
G	∞	-



	1	2	3	4	5	inf
B	○	○	○	○	○	○
C	○	○	○	○	○	○
D	○	○	○	○	○	○
E	○	○	○	○	○	○
F	○	○	○	○	○	○
G	○	○	○	○	○	○

True or False: Dijkstra's uses a First-In-First-Out queue to determine which vertex to visit next

0 points

- ○ True
- ○ False

The BART is coding a navigation app and needs your help. The subway system can be represented as a weighted graph, where edges represent the time it takes to ride between stations. Their app should be able to efficiently provide the quickest ride between a user's start & end stops. 2 points

For instance, if the user wants to go from Embarcadero to Berryessa, the app outputs: Embarcadero → OAK → Berryessa

Which algorithm should they use?

- Breadth First Search
- Depth First Search
- Dijkstra's
- A*

A copy of your responses will be emailed to yiyunchen@berkeley.edu.

Submit

Clear form