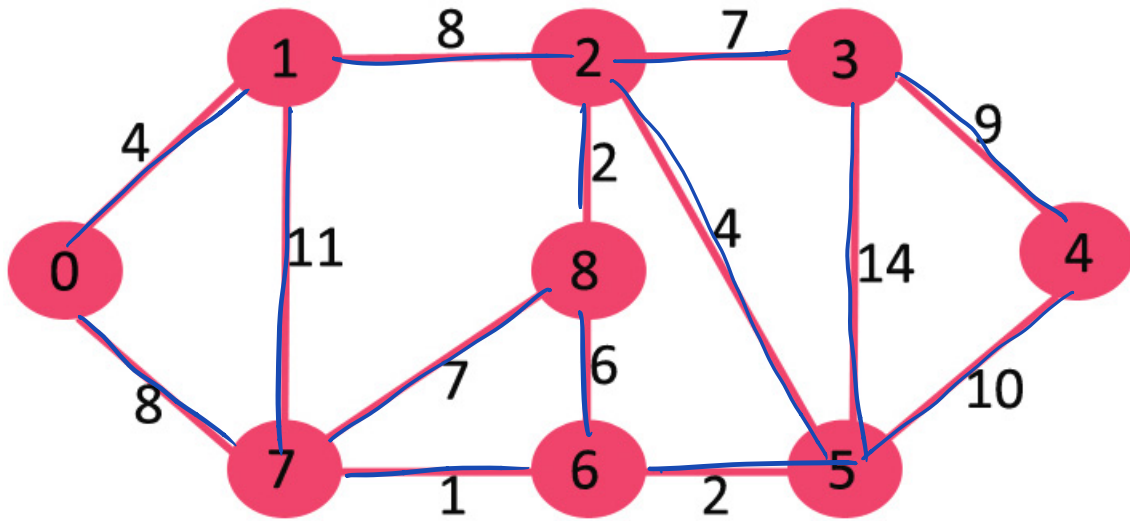


1 Graphs



The following questions use the same graph above unless explicitly specified. Some questions may have additional verbal modifications (without attaching a new graph).

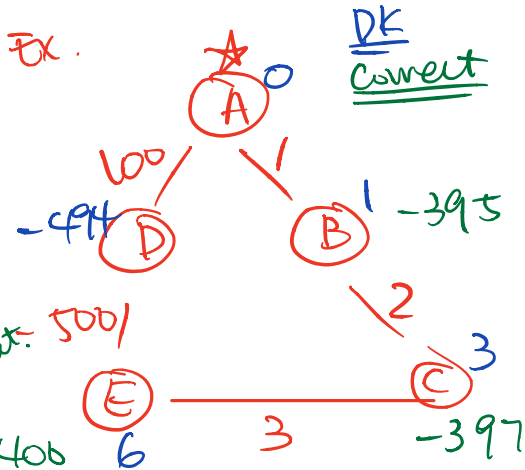
1.1 Dijkstra

1. Perform a Dijkstra algorithm from **Vertex 0**. Please indicate the final weight for each vertices and the corresponding path.
2. Give one graph and explain why Dijkstra doesn't work with negative edges.
3. Change one weight to negative above and show if the correctness is affected. Why? (Why it is wrong, or why it remains correct).

<i>dist</i>	<i>order</i>	<i>Vertices</i>	<i>Distance To []</i>	<i>Edge To []</i>
PQ (0, 0)	①	0 0	0	—
(1, 4)	②	1	4	0
(2, 12)	⑥	2	12	1
(3, 25)	⑧	3	25 19	2
(4, 21)	⑨	4	21	5
(5, 11)	⑤	5	11	6
(6, 9)	④	6	9	7
(7, 8)	③	7	8	0
(8, 15)	⑦	8	15 14	7 2

2. Two Reasons

- ① We don't check if certain vertices are visited when we relax them in Dijkstra. Therefore, if we look at all the adjacent nodes, the same node (aka the parent) will be affected again, back and forth, which creates an infinite loop.
- ② Even if we can't visit in Dijkstra, negative edges will remain troublesome because it would cause incorrect solution under some situations.



If we explore the left side first, but the distance to D is not correct. Thus, just don't use negative values for Dijkstra.

We start with A and Dijkstra will prefer the shorter edges on the right, and determine and confirm the distance based on that, while the -500 will actually create a much smaller weight.

3. Change any edges would cause incorrect answer if we don't check "visited" while relaxing. (Might update the parent or edge from incorrectly).

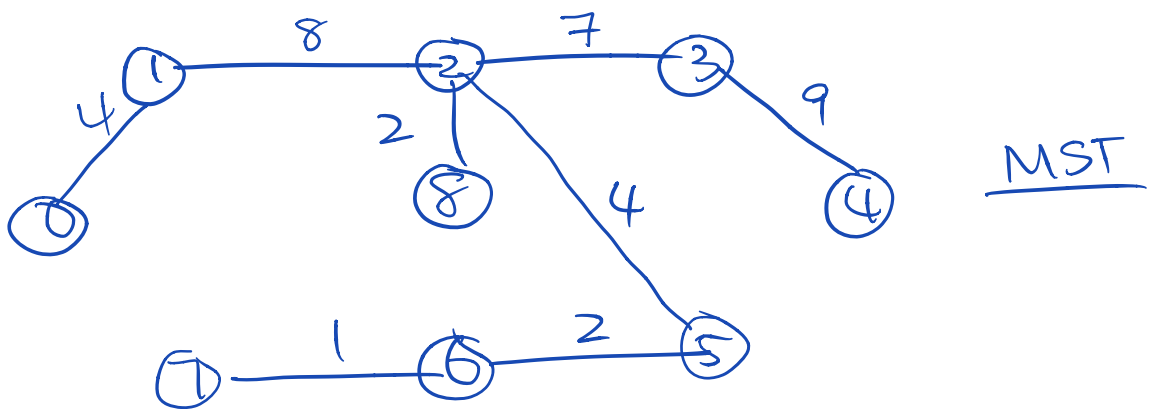
If we check existence in the PQ (aka not visited), edge 6-7 would not hide the correct solution because it's the shortest edge already. Answer is arbitrary,

1.2 Prim

- Using the original graph (without the modification in 1.1.3), find the minimum spanning tree using Prim Algorithm.
- Using the simple graph you constructed from 1.1.2, find the MST using Prim. Is it affected by negative weights? Why?
- Using the modified graph (1.1.3), find the MST using Prim.

vertex	weight	order	weight	Edges	visited
PQ (0, 0)	0	①	0	—	✓
(1, 4)	4	②	1	0	✓
(2, 8)	8	③	2	1	✓
(3, 7)	7	④	3	2	✓
(4, 10)	10	⑤	4	3	✓
(5, 4)	4	⑥	5	2	✓
(6, 2)	2	⑦	6	5	✓
(7, 8)	8	⑧	7	6	✓
(8, 2)	2	⑨	8	2	✓

Break Ties by alphabet, but didn't declare in problem.



For 2,3 since Prim only look at one edge (instead of total distance from origin), and un-visited is required, Prim is safe to have negative weights.