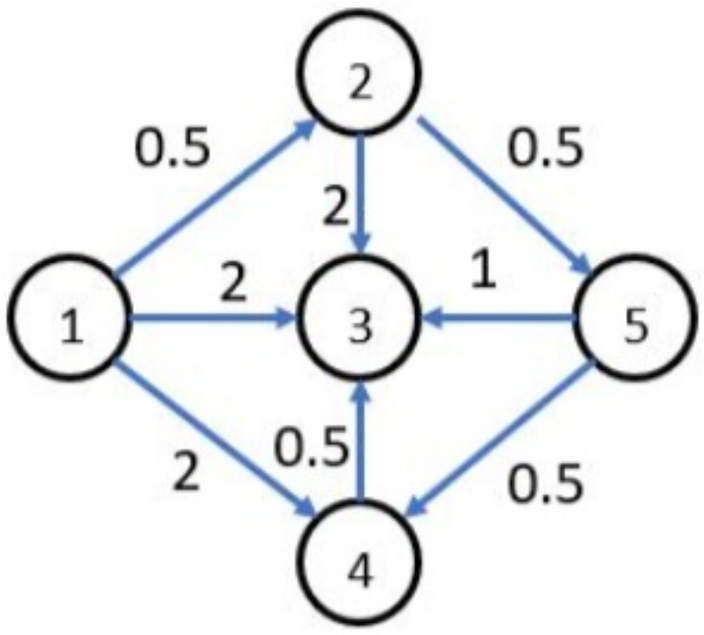
1. We will apply Bellman ford algorithm for the following graph with the source vertex 1.

1 / 1 point



Suppose the current distance estimates and the parents are :

Node	Distance (d)	Parent(pi)
1	0	NIL
2	0.5	1
3	2.5	4
4	2	1
5	infinity	NIL

- [x] Relaxing according to the edge 2-> 5 will bring down the weight of node 5 from infinity to 1 and set parent pointer of 5 to 2.

Correct
Correct.

- [x] The distance estimate of 2.5 for node 3 is the weight of the non-shortest path 1-> 4 -> 3

Correct
Correct – we see that the parent of 3 is 4 and the parent of 4 is 1.

- [x] Since there is no negative weight cycle in this graph running 4 iterations of relax, each iteration involving every edge in some order will guarantee that all shortest weight paths are found.

Correct
Correct.

- [ ] The shortest path distances are all final since no further relax operations will successfully decrease the distances of any vertex.
- [ ] If we relax according to the edge (5 , 3) the distance est. to node 3 is set to infinity as well.

2. Select all true facts about the Bellman Ford Algorithm.

1 / 1 point

- [x] If after one iteration that goes through and relaxes according to all the edges of the graph, we find that the distance estimates have not changed, then we can stop the algorithm (potentially early), concluding that the final weights have been found.

Correct
This is correct since further iterations will not modify any of the weights.

- [x] If after n-1 iterations where n is the number of vertices, the distance estimates are such that we can find edges that can successfully decrease them for some vertices, then there must be a negative weight cycle.

Correct
Correct – this was argued in the lecture and is formally proved in the book, as well.

- [x] The order in which the edges are iterated does not affect the correctness of Bellman-Ford but may affect the performance on some instances.

Correct
This is exactly right. We will have a chance to observe this further.

- [ ] Rather than iterate n-1 times through all the edges, suppose we flipped the order of loops in the algorithm:

  For e in Edges:

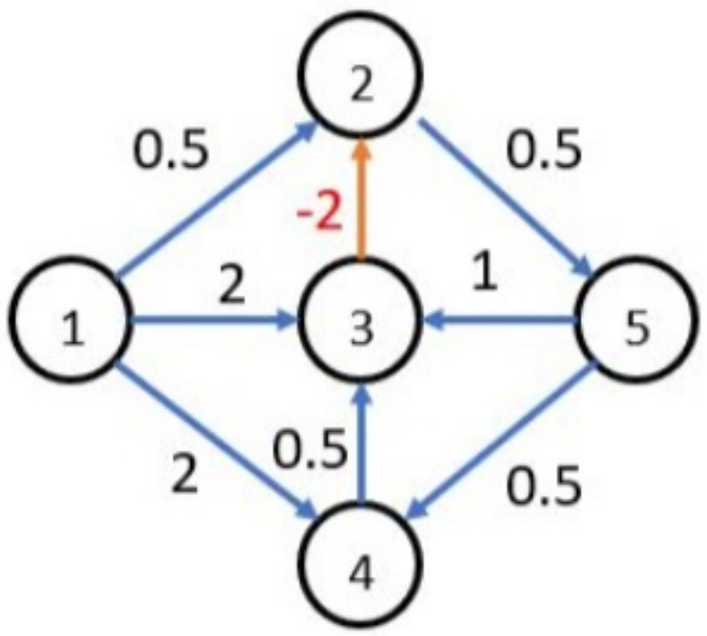
  For I from 1 to n-1:

  Relax(e)

  It continues to work just the same as before.

3. Consider the modified graph from previous question.

1 / 1 point



Suppose we have the distance estimates and parents computed after the first 4 iterations of Bellman-Ford algorithm for source vertex 1 (under some arbitrary order of relaxing the edges)

Select all the edges that can further relax the distance estimates shown:

Node	d	parent
1	0	NIL
2	-1	3
3	0.5	5
4	0	5
5	-0.5	2

- [ ] (1,2)
- [ ] (1,4)
- [ ] (5,3)
- [x] (3,2)

Correct

Correct. Distance estimate to 3 is 0.5, weight of edge is -2, therefore it can update distance to 2 to -1.5