1. # We will apply Dijkstra's algorithm for the following graph with the source vertex 1.
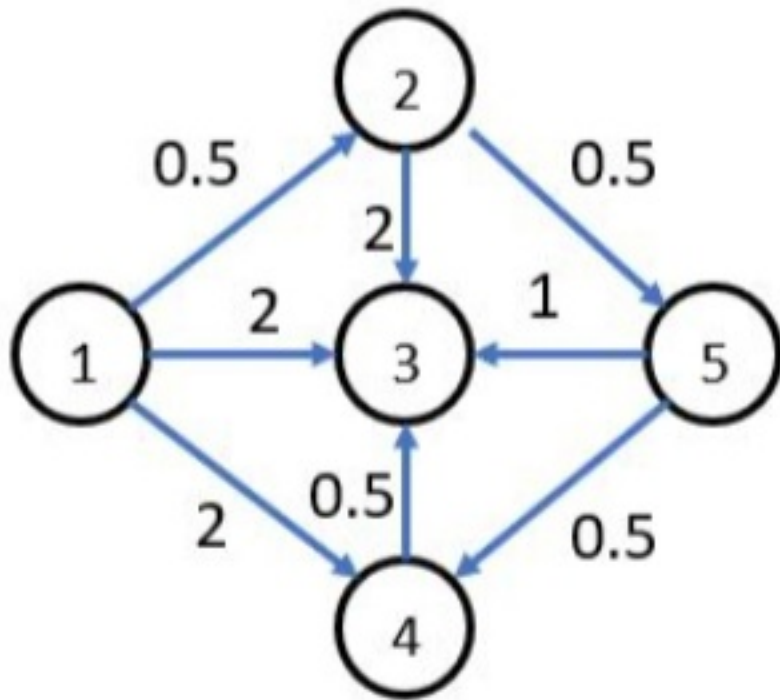

Here is the table of nodes with distances and parent values. Nodes 1 and 2 have already been visited or processed.

Node	Distance	Parent
1	0	NIL
2	0.5	1
3	2	1
4	2	1
5	1	2

Select all the correct facts about the subsequent working of the algorithm.

☐ The next node to be visited will be node 3.

☑ Dijkstra's algorithm will relax each edge exactly once during its entire execution.

> ⊘ **Correct**
> Correct. Specifically, an edge (u,v) is relaxed when its source node u is popped off the priority queue.

☐ The lowest priority node is node 1 and will be visited next.

☑ The priority queue (heap) has nodes 3, 4 and 5 with priorities 2, 2 and 1 respectively.

> ⊘ **Correct**
> That is correct. Unvisited/unprocessed nodes will be in the priority queue with the distance estimates as the priority.

☑ The next node to be visited will be 5 and will cause edges (5,4) and (5,3) to be relaxed.

> ⊘ **Correct**
> Correct.

2. In this question, we will examine some of the properties that are true of Dijkstra's algorithm in general. Here is a schematic of a graph representing an intermediate stage of running the algorithm.
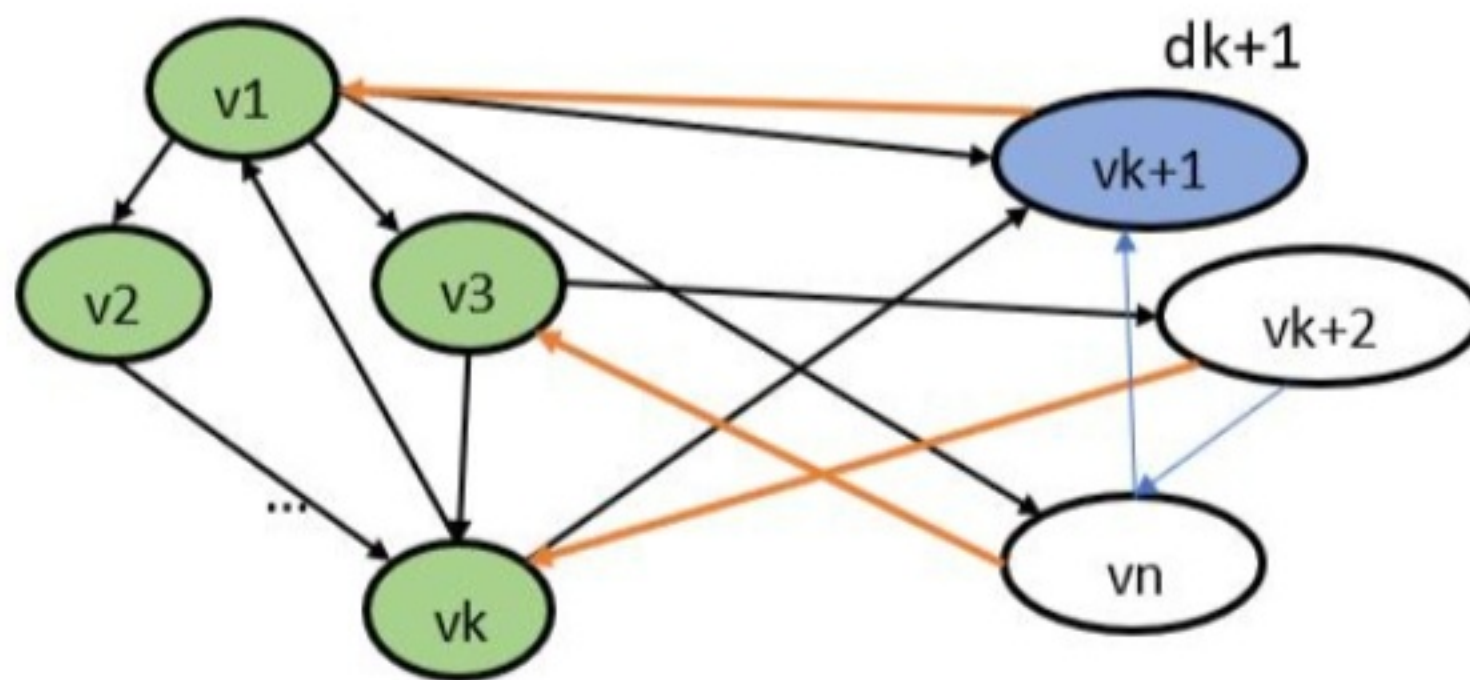
The green nodes $v_1, \ldots, v_k$ represent nodes already visited or processed. The other nodes $v_{k+1}, \ldots, v_n$ represent the as yet unprocessed nodes with v_{k+1} representing the node with lowest priority given by d_{k+1} (assume all other unvisited nodes have strictly larger priority). Select all the true facts about properties of Dijkstra's algorithm from the list below.

- [ ] A previously visited node can be added back to the priority queue as the algorithm runs.

- [x] The shortest path to node v_{k+1} must have shortest path distance d_{k+1}.

> ⊘ **Correct**
> Correct, as we argued in the lecture.

- [ ] It is possible that when an outgoing edge from v_{k+1} is relaxed in the subsequent step, the distance estimate to some green previously visited node decreases.

- [x] The next node that will be visited by Dijkstra's algorithm is v_{k+1}.

> ⊘ **Correct**
> Correct.

- [x] For all visited nodes (green nodes in graph), the distance estimate is less than or equal to d_{k+1}.

> ⊘ **Correct**
> Correct. This is because all edge weights are non-negative.