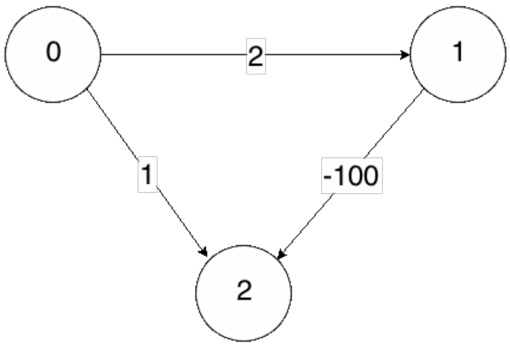
15210: Parallel and Sequential Data Structures and Algorithms

AbridgedLab

Zikang Wang (zikangw)

5.1



Node	PQ Operation	PQ State
0	insert(1, 2), insert(2, 1)	{2→1, 1→2}
2	-	{1→2}

From 0 to 2, there are 2 paths, and 0→1→2 is shortest (-98). But When using dijkstra, we would pick 0→2, regard this as the shortest path, and never update it again.

To fix this, we should update all the distances to all the vertices on frontier each time. In this way, in case of there is a path with negative weight, we can identify it when update.

5.2

Disagree.

It is possible that in one round we access the same node from multiple frontier nodes. In this case we would visit this node twice. Therefore, even if we only visit unvisited nodes, we can't guarantee O(n) work.

5.3

1) Admissible:

Euclidean distance between two nodes is the minimal possible distance between the nodes.

Therefore, h(v) is always less than or equal to the shortest distance d(v) (equal when shortest path is a straight line), and the heuristic is admissible.

2) Consistent:

When heuristic is Euclidean distance, h(u), h(v) and e(u, v) form a triangle. In a triangle, sum of any two edges is larger than the third edge, therefore,

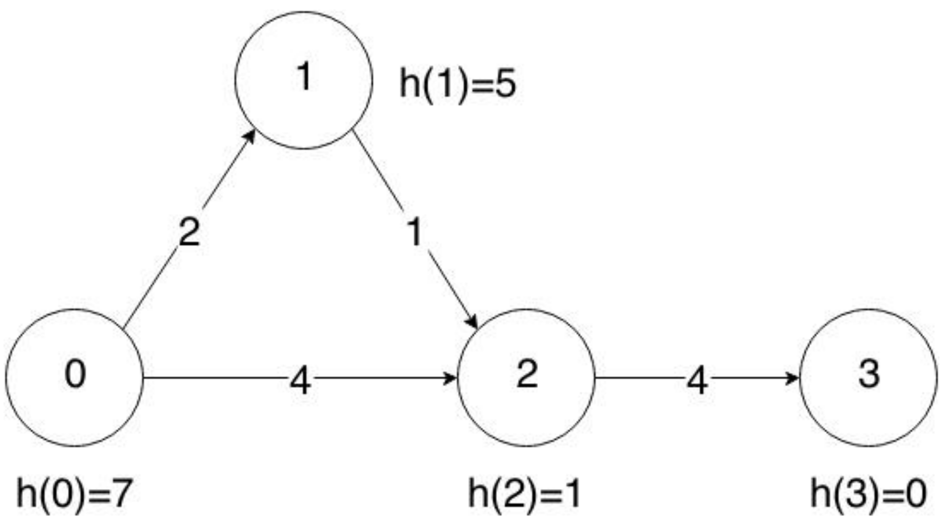
$$h(u) \leq h(v) + e(u, v)$$

Thus, it is consistent.

5.4

Set heuristic to 0 for all vertices (or any other value, as long as h for all vertices are the same), then it would act as dijkstra.

5.5



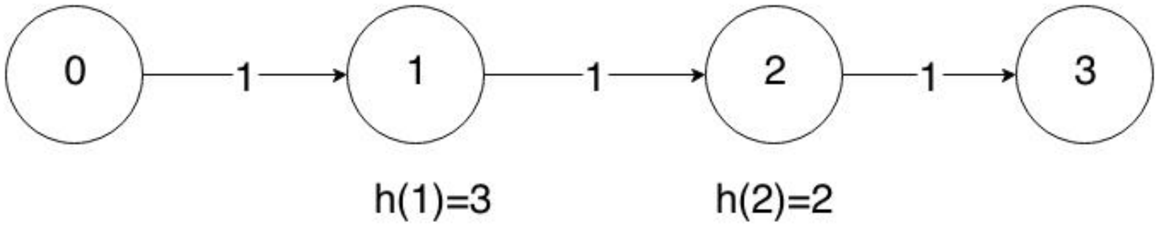
As shown above, destination is vertex 3.

	0	1	2	3
Heuristic	7	5	1	0
Shortest Dist	8	5	4	0

We can see that for each vertex $heuristic \leq shortest\ distance\ to$ 3, therefore it is admissible.

But for vertex 1 and 2, $h(1) > h(2) + w(1, 2)$, not consistent.

5.6



As shown above,

1: $h(1) = 3 > min\_dist(1) = 2$

2: $h(2) = 2 > min\_dist(2) = 1$

Therefore it is not admissible.