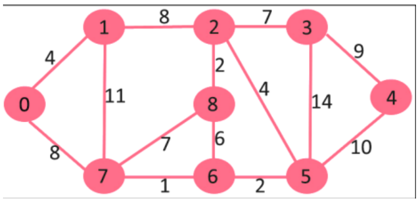
Single Source Shortest Path: Dijkstra's algo



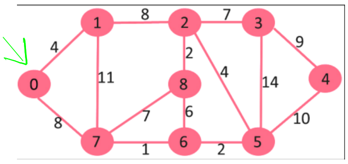
- Dijkstra's Algorithm will work for both Directed and undirected graphs but only when the weights are positive!

Procedure:

- Step 01: Construct the cost adjacency matrix for the given graph.
- Step 02: Assume a vertex as the source (alphabetically) (if source is not mentioned) and compute the distance from the source to all other vertices as D[w]=c(s,w) or c(s,u)+c(u,w) i.e. direct distance or indirect distance. This is also known as relaxation.
- Step 03: Pick the shortest path of the computed distance.
- Step 04: The vertex causing the shorted path is also included into the source.
- Step 05: Repeat the steps till all shortest paths are evaluated.

adjacency matrix ADM

to/from	0	1	2	3	4	5	6	7	8
0	inf	4	inf	inf	inf	inf	inf	8	inf
1	4	inf	8	inf	inf	inf	inf	11	inf
2	inf	8	inf	7	inf	4	inf	inf	2
3	inf	inf	7	inf	9	14	inf	inf	inf
4	inf	inf	inf	9	inf	10	inf	inf	inf
5	inf	inf	4	14	10	inf	2	inf	inf
6	inf	inf	inf	inf	inf	2	inf	1	6
7	8	11	inf	inf	inf	inf	1	inf	7
8	inf	inf	2	inf	inf	inf	6	7	inf

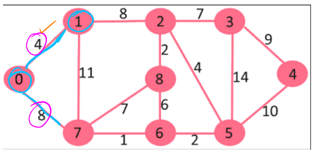


0 1 2 3 4 5 6 7 8
∞ ∞ ∞ ∞ ∞ ∞ ∞ ∞

PrQ = [(0,0)] {stores (dist, node)}

PrQ-GET

Visited : [0], Unvisited : [1,2,3,4,5,6,7,8]



0 1 2 3 4 5 6 7 8
0 4 ∞ ∞ ∞ ∞ ∞ 8 ∞

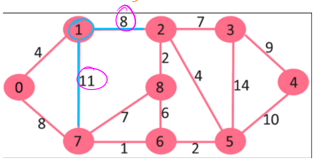
add all nodes that are reachable from current node into PrQ with dist = dist till current node + dist from current node to reachable node

PrQ = [(4,1) (8,7)]

current dist

PrQ-GET

Visited : [0,1], Unvisited : [2,3,4,5,6,7,8]



0 1 2 3 4 5 6 7 8
0 - 4 ∞ ∞ ∞ ∞ ∞ 8 ∞
1 12 ∞ ∞ ∞ ∞ 15 ∞

4+8=ADM[1][2], 4+11=ADM[1][7]

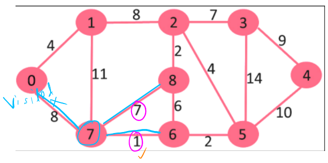
update new distance as { Min distance till current node + Dist from current to next reachable node from adjacency matrix }

PrQ = [(8,7), (12,2)]

current dist

PrQ-GET

Visited : [0,1,7], Unvisited : [2,3,4,5,6,8]



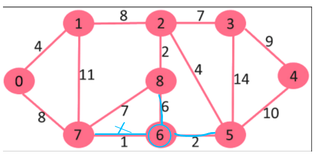
0 1 2 3 4 5 6 7 8
0 - 4 ∞ ∞ ∞ ∞ ∞ 8 ∞
1 12 ∞ ∞ ∞ ∞ 15 ∞
7 ∞ ∞ ∞ ∞ 9 ∞ 15

8+1, 8+7

PrQ = [(9,6), (12,2), (15,8)]

current dist

Visited : [0,1,7,6], Unvisited : [2,3,4,5,8]



0 1 2 3 4 5 6 7 8
0 - 4 ∞ ∞ ∞ ∞ ∞ 8 ∞
1 12 ∞ ∞ ∞ ∞ 15 ∞
7 - ∞ ∞ ∞ ∞ 9 ∞ 15
6 ∞ ∞ ∞ 11 ∞ - 15

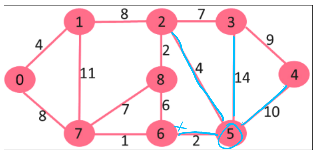
9+2, 9+6

PrQ = [(11,5), (12,2), (15,8)]

PrQ-GET 5

current dist

Visited : [0,1,7,6,5], Unvisited : [2,3,4,8]



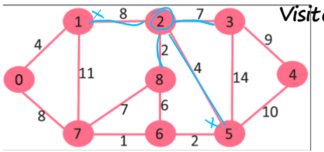
0 1 2 3 4 5 6 7 8
0 - 4 ∞ ∞ ∞ ∞ ∞ 8 ∞
1 - 12 ∞ ∞ ∞ ∞ 15 ∞
7 - ∞ ∞ ∞ ∞ 9 - 15
6 ∞ ∞ ∞ 11 ∞ - 15
5 15 25 21 - ∞ ∞ ∞

11+4, 11+14, 11+10

since one shortest path to 2 is already in PrQ ∴ didn't added (15,2)

PrQ = [(12,2), (15,8), (21,4), (25,3)]

Visited : [0,1,7,6,5,2], Unvisited : [3,4,8]



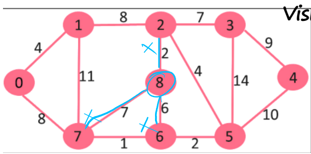
0 1 2 3 4 5 6 7 8
0 - 4 ∞ ∞ ∞ ∞ ∞ 8 ∞
1 - 12 ∞ ∞ ∞ ∞ 15 ∞
7 - ∞ ∞ ∞ ∞ 9 - 15
6 ∞ ∞ ∞ 11 ∞ - 15
5 15 25 21 - ∞ ∞ ∞
2 - 19 ∞ 16 ∞ ∞ 14

12+7, 12+4, 12+2

new path to 8 from 2 has dist 14 which is less than already discovered dist to 8 i.e 15 so adding (14,8) to PrQ

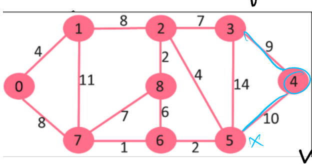
PrQ = [(14,8), (15,8) (19,3), (21,4), (25,3)]

Visited : [0,1,7,6,5,2,8], Unvisited : [3,4]



0 1 2 3 4 5 6 7 8
0 - 4 ∞ ∞ ∞ ∞ ∞ 8 ∞
1 - 12 ∞ ∞ ∞ ∞ 15 ∞
7 - ∞ ∞ ∞ ∞ 9 - 15
6 ∞ ∞ ∞ 11 ∞ - 15
5 15 25 21 - ∞ ∞ ∞
2 - 19 ∞ 16 ∞ ∞ 14
8 - - ∞ ∞ ∞ 20 21 -

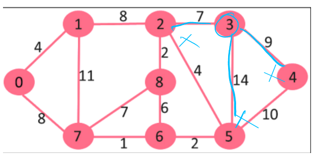
existing shortest dist to 8 is from 2



PrQ = [(21,4), (25,3)]

Visited : [0,1,7,6,5,2,8,4], Unvisited : [3]

0 1 2 3 4 5 6 7 8
4 - ∞ ∞ 30 - - ∞ ∞ ∞



PrQ = [(25,3)]

Visited : [0,1,7,6,5,2,8,4,3], Unvisited : []

stop

Links:

Priotiy Queue: https://www.youtube.com/watch?v=wptevkObshY

Node	0	1	2	3	4	5	6	7	8
Dist	0	4	12	19	21	11	9	8	14
Prev	0	0	1	2	5	6	7	0	2