Exp 5

SSSP Problem

1] Dijkstra's Algo.

* Graph

Vertices : $\{0, 1, 2, 3\}$

Edges: $0 \rightarrow 1$ (weight 4)
$0 \rightarrow 2$ (1)
$2 \rightarrow 1$ (2)
$1 \rightarrow 3$ (1)
$2 \rightarrow 3$ (5)

* Execution

1.) Initializatⁿ

dist = $[0, \infty, \infty, \infty]$ ..... src 0

pq = $[(0,0)]$ ...... distance 0 to vertex 0

2.) First iteratⁿ

process vertex 0 (smallest dist. in pq)

relax edges from 0

Edge $0 \rightarrow 1$: update dist[1] = 0 + 4 = 4

$0 \rightarrow 2$: " dist[2] = 0 + 1 = 1

pq = $\{(1,2), (4,1)\}$

3.) Second it.

process vertex 2 (relax edges from 2)..... smallest dist. in pq

$2 \rightarrow 1$: update dist[1] = 1 + 2 = 3 (shorter than 4)

$2 \rightarrow 3$: update dist[3] = 1 + 5 = 6

pq = $\{(3,1), (6,3)\}$

4.) Third it.

process vertex 1 (smallest dist in pq)

relax edges from 1

$1 \rightarrow 3$: update dist[3] = 3 + 1 = 4 (shorter than 6)

pq = $\{(4,3)\}$

5.) Fourth it.

process vertex 3 (smallest dist. in pq)

no edges to relax.

pq = { }

6.) Algo. terminates when pq is empty

final 'dist' arr = [0, 3, 1, 4]

2.] Bellman-Ford Algo.

* Graph

Vertices: {0, 1, 2, 3}

Edges: 0 → 1 (wt. 4)

0 → 2 (1)

2 → 1 (−2)

1 → 3 (1)

2 → 3 (5)

* Executⁿ

1.) Init.

dist = [0, ∞, ∞, ∞] ...... src. 0

2.) Relaxatⁿ (v − 1 times = 3 times)

i.) 1st iteratⁿ: − Relax all edges

0 → 1 : update dist[1] = 0 + 4 = 4

0 → 2 : " dist[2] = 0 + 1 = 1

2 → 1 : " dist[1] = 1 + (−2) = −1

1 → 3 : " dist[3] = −1 + 1 = 0

2 → 3 : " dist[3] = 0 .... not updated ∵ 0 < 6

1+5

dist = [0, −1, 1, 0]

ii) 2nd it. − Relax all edges

No further updates

iii.) 3rd it. − Relax all edges

No further updates

3] All Pairs Shortest Path

* Graph

Vertices : {0, 1, 2}

Edges: $0 \to 1$ (wt. 2)

$0 \to 2$ (4)

$1 \to 2$ (1)

$2 \to 0$ (1)

* Executⁿ - For each vertex as src, run Dijkstra's algo. & return 2D arr. with shortest distances.

1.) Src. 0

$0 \to 0$: dist[0] = 0 .... src 0

$0 \to 1$: dist[1] = 2

$0 \to 2$: path $0 \to 1 \to 2$, dist[2] = 2 + 1 = 3

2.) Src. 1

1 to 0 : no direct path ; dist[0] = INF

1 to 1 : dist[1] = 0 ..... src 1

1 to 2 : dist[2] = 1

3.) Src. 2

2 to 0 : dist[0] = 1

2 to 1 : Path $2 \to 0 \to 1$, dist[1] = 1 + 2 = 3

2 to 2 : dist[2] = 0 ..... src 2

'allDistances' arr = $\begin{bmatrix} 0 & 2 & 3 \\ \infty & 0 & 1 \\ 1 & 3 & 0 \end{bmatrix}$

* This example depicts limitatⁿ of Dijkstra's algo. - it's a greedy algo. that assumes that once a vertex is processed, its shortest dist. from the src. is finalized.

* Execut^{n} - For each vertex run Bellman-Ford algo.

1.) Src. 0

0 to 0 : dist[0] = 0

0 to 1 : dist[1] = 2

0 to 2 : $0 \to 1 \to 2$, dist[2] = 2+1 = 3

2.) Src. 1

1 to 0 : $1 \to 2 \to 0$, dist[0] = 1+1 = 2

1 to 2 : dist[2] = 1

1 to 1 : dist[1] = 0

3.) Src. 2

2 to 0 : dist[0] = 1

2 to 1 : $2 \to 0 \to 1$, dist[1] = 1+2 = 3

2 to 2 : dist[2] = 0

'allDistances' arr =
$$\begin{bmatrix} 0 & 2 & 3 \\ 2 & 0 & 1 \\ 1 & 3 & 0 \end{bmatrix}$$

- Correctly computes shortest paths in graphs with cycles and negative edge weights.
- Detects -ve ~~edge~~ weight cycles if they exist.