

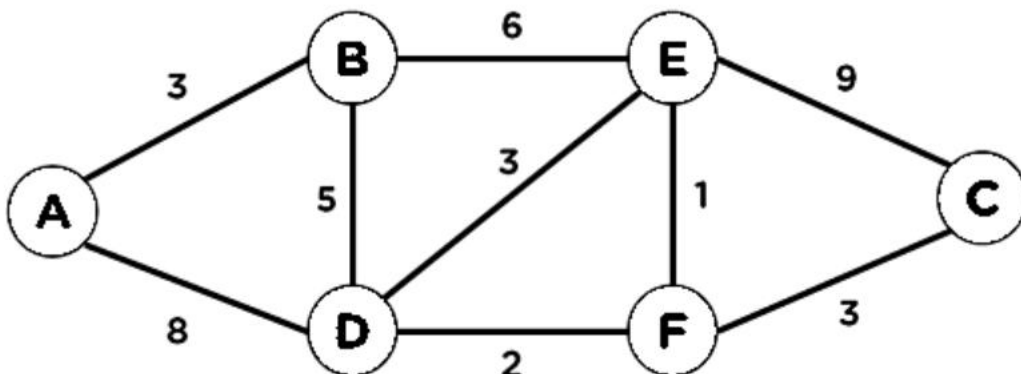
Course/ Batch: BTech/ SCSET
Course Code: CSET244
Semester: Even
Session: 2024-2025

Course Type: Core
Course Name: Design and Analysis of Algorithms

Tutorial Assignment: Week 8

Tutorial title: Dijkstra's Algorithm

Q1: Use Dijkstra's algorithm as greedy approach to determine shortest distance from node A to C.



Solution:

1. All the distances from node A to the rest of the nodes is ∞ .
2. Calculating the distance between node A and the immediate nodes (node B & node D):
For node B,
Node A to Node B = 3
For node D
Node A to Node D = 8
3. Choose the node with the shortest distance to be the current node from unvisited nodes, i.e., node B.
Calculating the distance between node B and the immediate nodes:
For node E,
Node B to Node D = $3+5 = 8$
For node E,
Node B to Node E = $3+6 = 9$
4. Choose the node with the shortest distance to be the current node from unvisited nodes, i.e., node D.
Calculating the distance between node D and the immediate nodes:
For node E,

Node D to Node E = $8+3 = 11$ ($[9 < 11] > \text{TRUE}$: So, No Change)

For node F,

Node D to Node F = $8+2 = 10$

5. Choose the node with the shortest distance to be the current node from unvisited nodes, i.e., node E.

Calculating the distance between node E and the immediate nodes:

For node C,

Node E to Node C = $9+9 = 18$

For node F,

Node E to Node F = $9+1 = 10$

6. Choose the node with the shortest distance to be the current node from unvisited nodes, i.e., node F.

Calculating the distance between node F and the immediate nodes:

For node C,

Node F to Node C = $10+3 = 13$ ($[18 < 13] \text{ FALSE}$: So, Change the previous value)

So, after performing all the steps, we have the shortest path from node A to node C, i.e., a value of 13 units

Q2. Consider the following directed graph

$A \rightarrow B$ (4)

$A \rightarrow C$ (0)

$C \rightarrow D$ (0)

$D \rightarrow E$ (5)

$B \rightarrow E$ (2)

If we apply Dijkstra's algorithm starting from A, what will be the shortest distance to E?

A) 4

B) 2

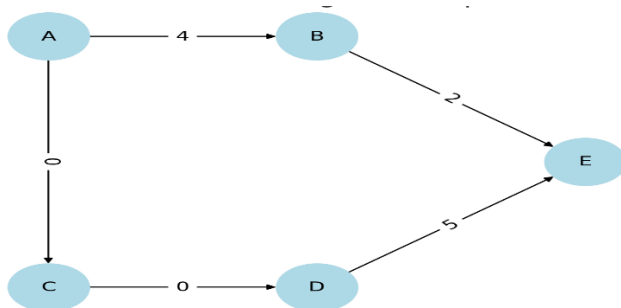
C) 5

D) 6

Solution

1. Answer: C) 5

Solution:



Initialize:

$\text{dist}(A) = 0, \text{dist}(B) = \infty, \text{dist}(C) = \infty, \text{dist}(D) = \infty, \text{dist}(E) = \infty$

Pick A (smallest distance = 0):

$A \rightarrow B$ (dist = 4)

$A \rightarrow C$ (dist = 0)

Pick C (smallest distance = 0):

$C \rightarrow D$ (dist = 0)

Pick D (smallest distance = 0):

$D \rightarrow E$ (dist = 5)

Pick B (smallest distance = 4):

$B \rightarrow E$ (new dist = $4 + 2 = 6$), but E is already at 5, so no update.

Final shortest distance to E = 5.

Trick: Zero-weight edges do not break Dijkstra's algorithm, but negative weights do.

Q3. Dijkstra's Algorithm maintains a parent array to reconstruct the shortest path from source to destination.
Given the following graph:

$A \rightarrow B$ (2)

$A \rightarrow C$ (4)

$B \rightarrow C$ (1)

$B \rightarrow D$ (7)

$C \rightarrow D$ (3)

$D \rightarrow E$ (1)

If we run Dijkstra's Algorithm from A to E, what is the shortest path and its total cost?

A) $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$, cost = 7

B) $A \rightarrow C \rightarrow D \rightarrow E$, cost = 8

C) $A \rightarrow B \rightarrow D \rightarrow E$, cost = 10

D) $A \rightarrow C \rightarrow E$, cost = 6

Solution: A) $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$, cost = 7

Initialize: $\text{dist}(A) = 0$, all others ∞

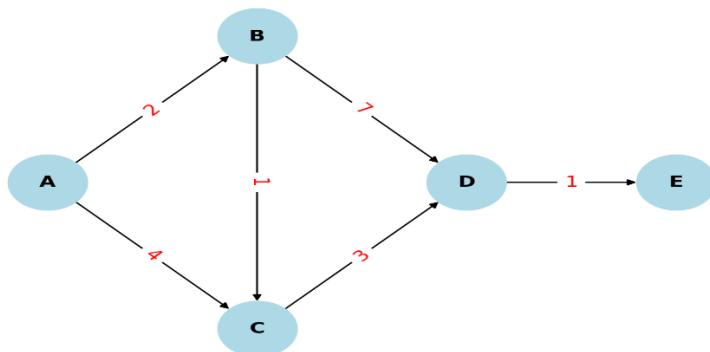
Pick A:

$A \rightarrow B$ (2), $A \rightarrow C$ (4)

Pick B (dist = 2):

$B \rightarrow C$ (new dist = $2 + 1 = 3$, update C)

$B \rightarrow D$ (new dist = $2 + 7 = 9$)



Q4. A directed graph contains some edges with zero weight. In order to ensure correctness, a transformation is applied before running Dijkstra's Algorithm. What transformation ensures correct results?

- (A) Replace all zero-weight edges with a very small positive value (e.g., 10^{-6}).
- (B) Treat zero-weight edges as unweighted and apply BFS instead.
- (C) Convert all zero-weight edges to positive weight edges by adding a constant to all edges.
- (D) Use Bellman-Ford Algorithm instead.

Solution: (A)

Zero-weight edges do not break Dijkstra's Algorithm, but in some cases, it may cause confusion in implementation. Adding a very small positive value ensures that the priority queue maintains correct behavior without significantly altering the results.

Note: Bellman-Ford is unnecessary unless negative weights exist.

Q5. Find the shortest path from A to all other vertices using the adjacency matrix for the graph using Dijkstra's Algorithm

	A	B	C	D	E
A	0	10	3	∞	∞
B	∞	0	1	2	∞
C	∞	4	0	8	2
D	∞	∞	∞	0	7
E	∞	∞	∞	9	0

Solution:

Final Shortest Distances:

Vertex	Distance from A
A	0
B	7
C	3
D	9
E	5