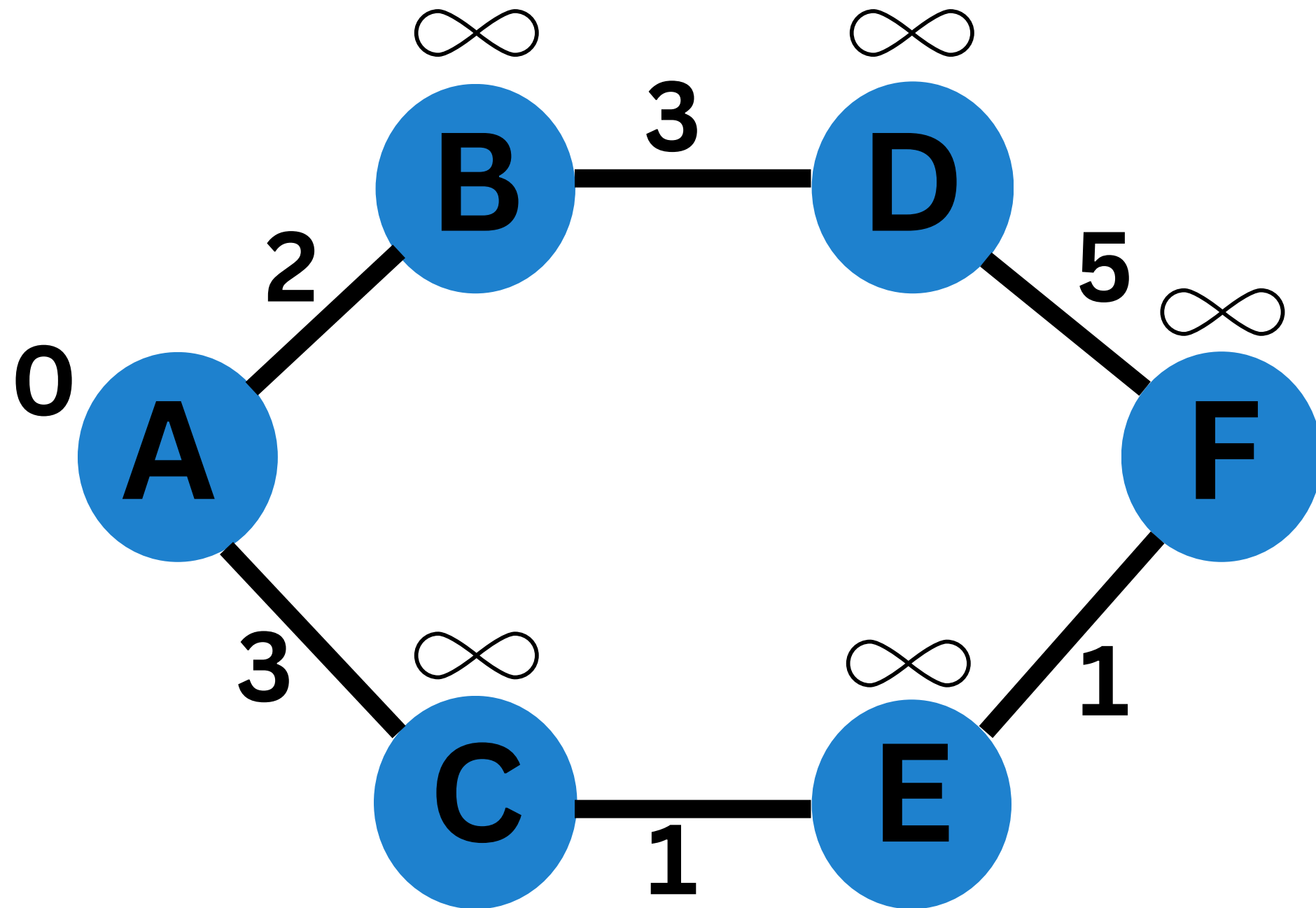


**THIS IS
DIJKSTRA**

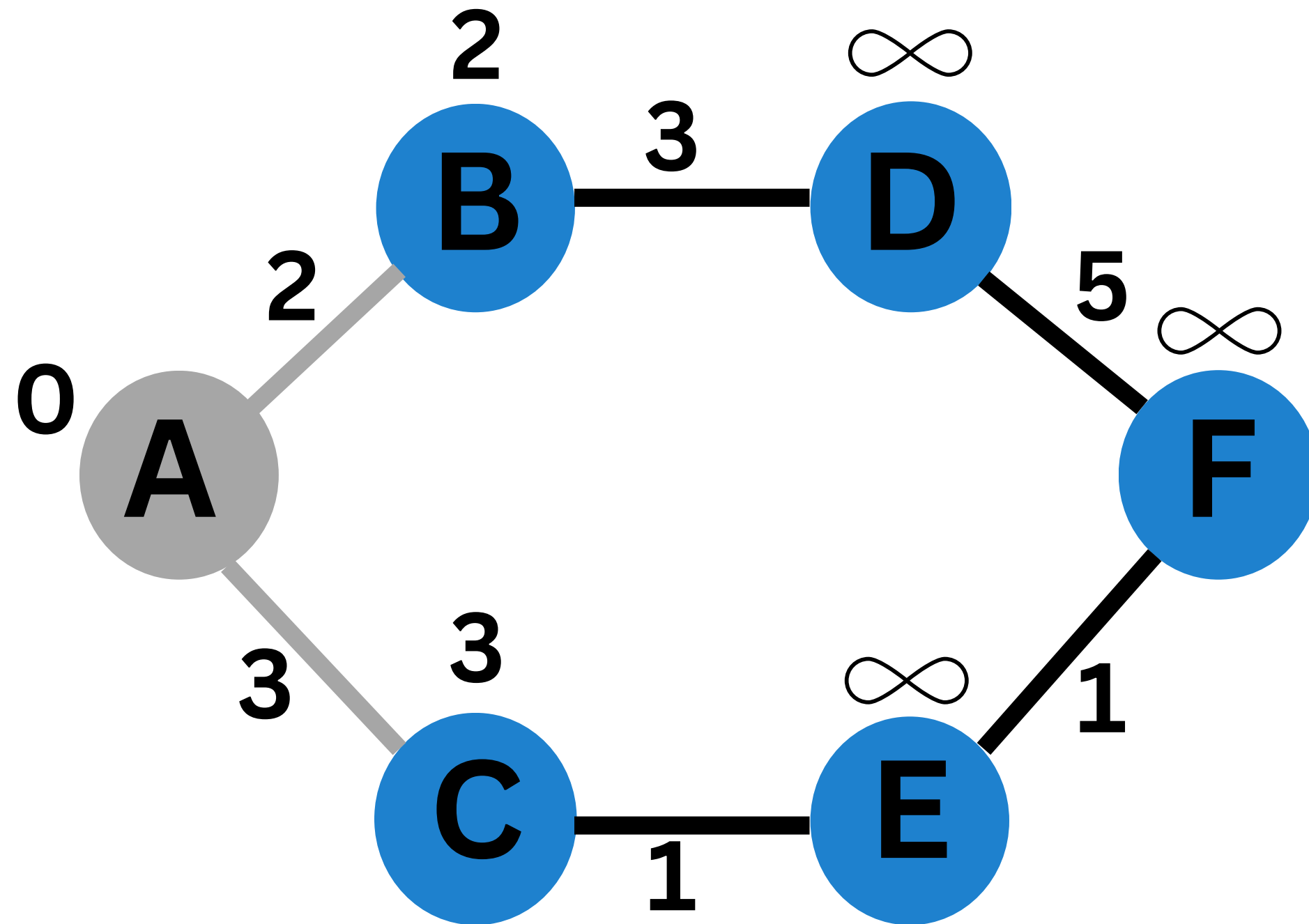
DIJKSTRA'S ALGORITHM



NODE	SHORTEST DISTANCE	PREVIOUS NODE
A	0	
B	∞	
C	∞	
D	∞	
E	∞	
F	∞	

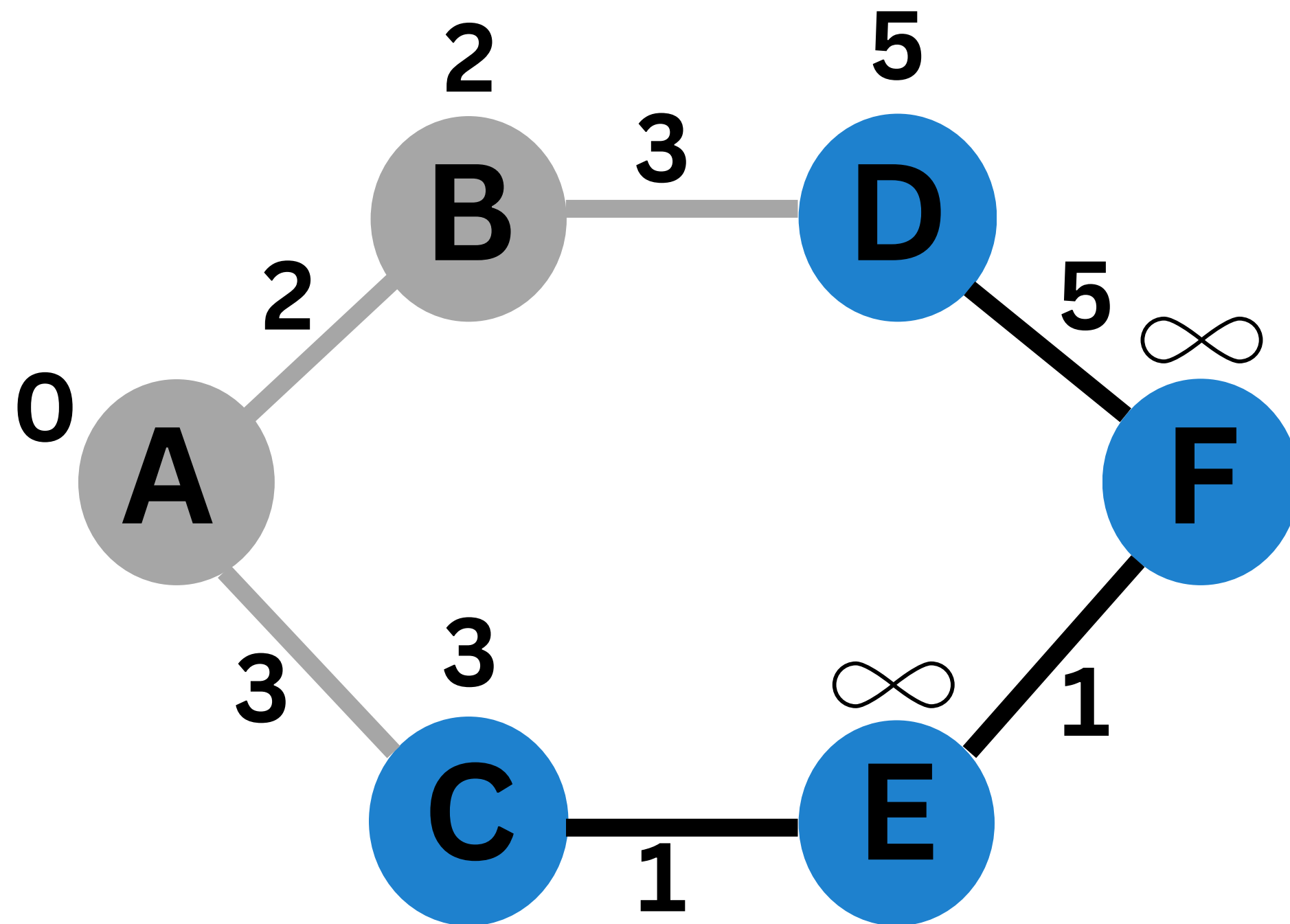
```
1  function Dijkstra(Graph, source):
2
3      for each vertex v in Graph.Vertices:
4          dist[v] ← INFINITY
5          prev[v] ← UNDEFINED
6          add v to Q
7      dist[source] ← 0
8
9      while Q is not empty:
10         u ← vertex in Q with min dist[u]
11         remove u from Q
12
13         for each neighbor v of u still in Q:
14             alt ← dist[u] + Graph.Edges(u, v)
15             if alt < dist[v]:
16                 dist[v] ← alt
17                 prev[v] ← u
18
19     return dist[], prev[]
```

DIJKSTRA'S ALGORITHM



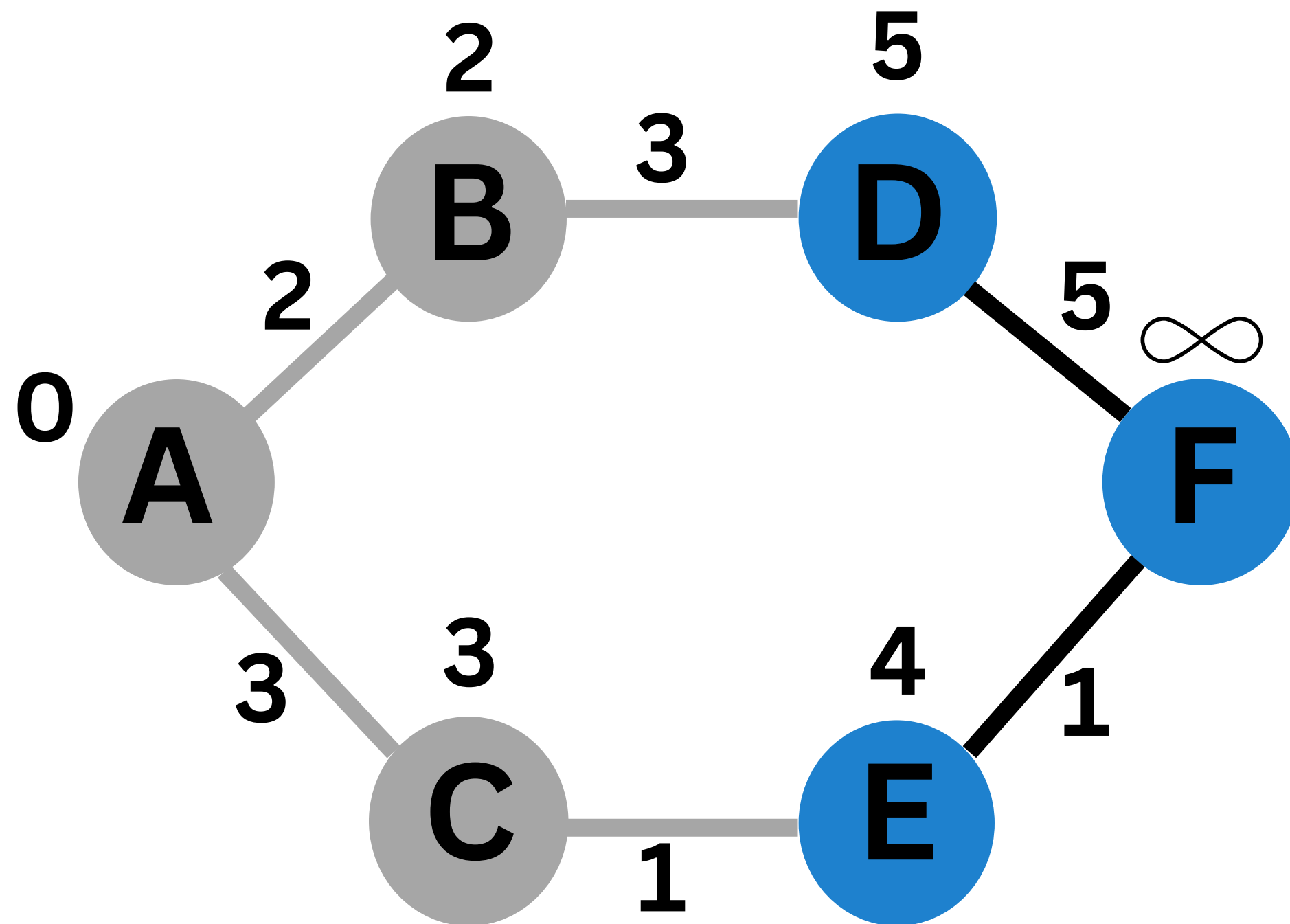
NODE	SHORTEST DISTANCE	PREVIOUS NODE
A	0	
B	2	A
C	3	A
D	∞	
E	∞	
F	∞	

DIJKSTRA'S ALGORITHM



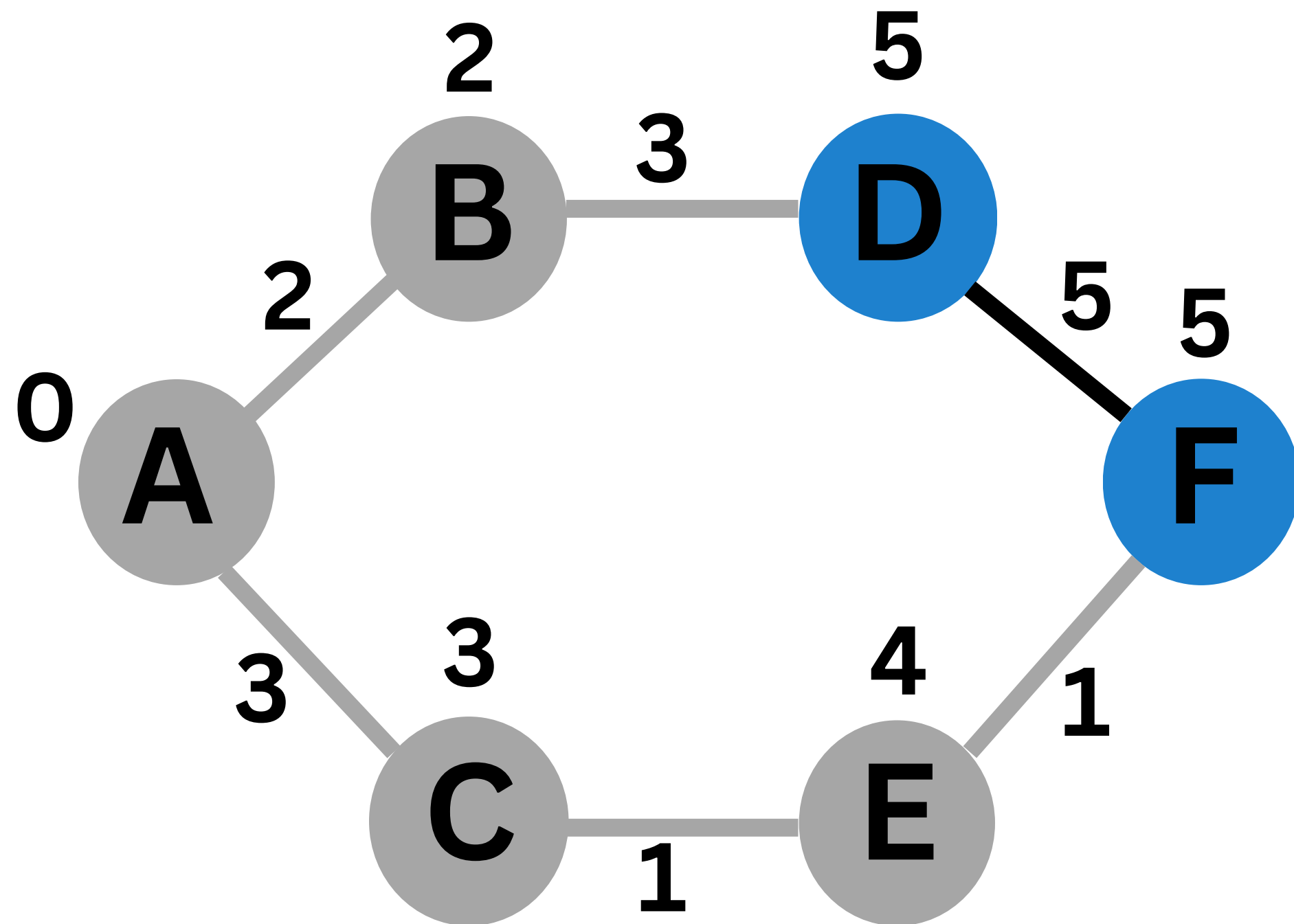
NODE	SHORTEST DISTANCE	PREVIOUS NODE
A	0	
B	2	A
C	3	A
D	5	B
E	∞	
F	∞	

DIJKSTRA'S ALGORITHM



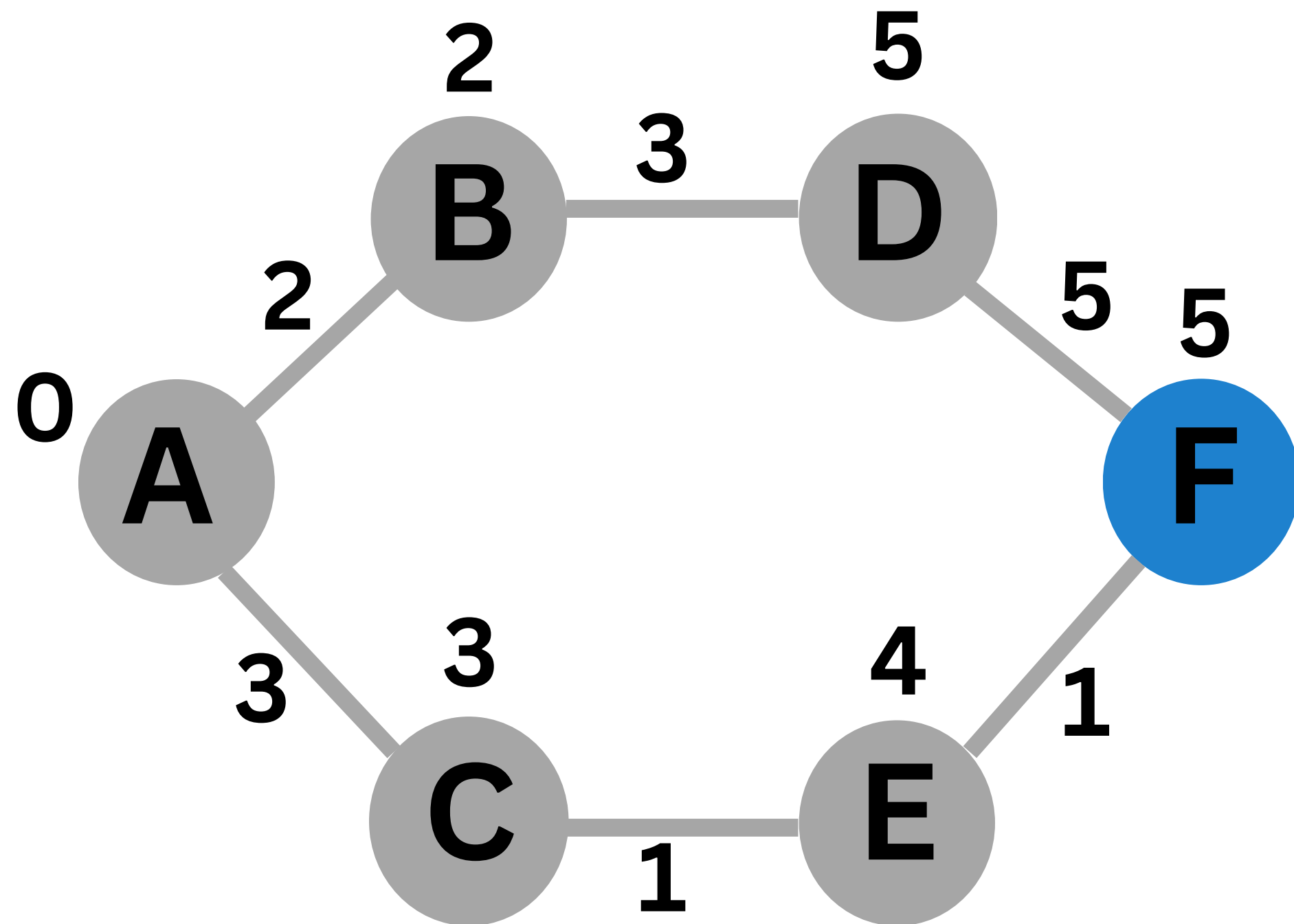
NODE	SHORTEST DISTANCE	PREVIOUS NODE
A	0	
B	2	A
C	3	A
D	5	B
E	4	C
F	∞	

DIJKSTRA'S ALGORITHM



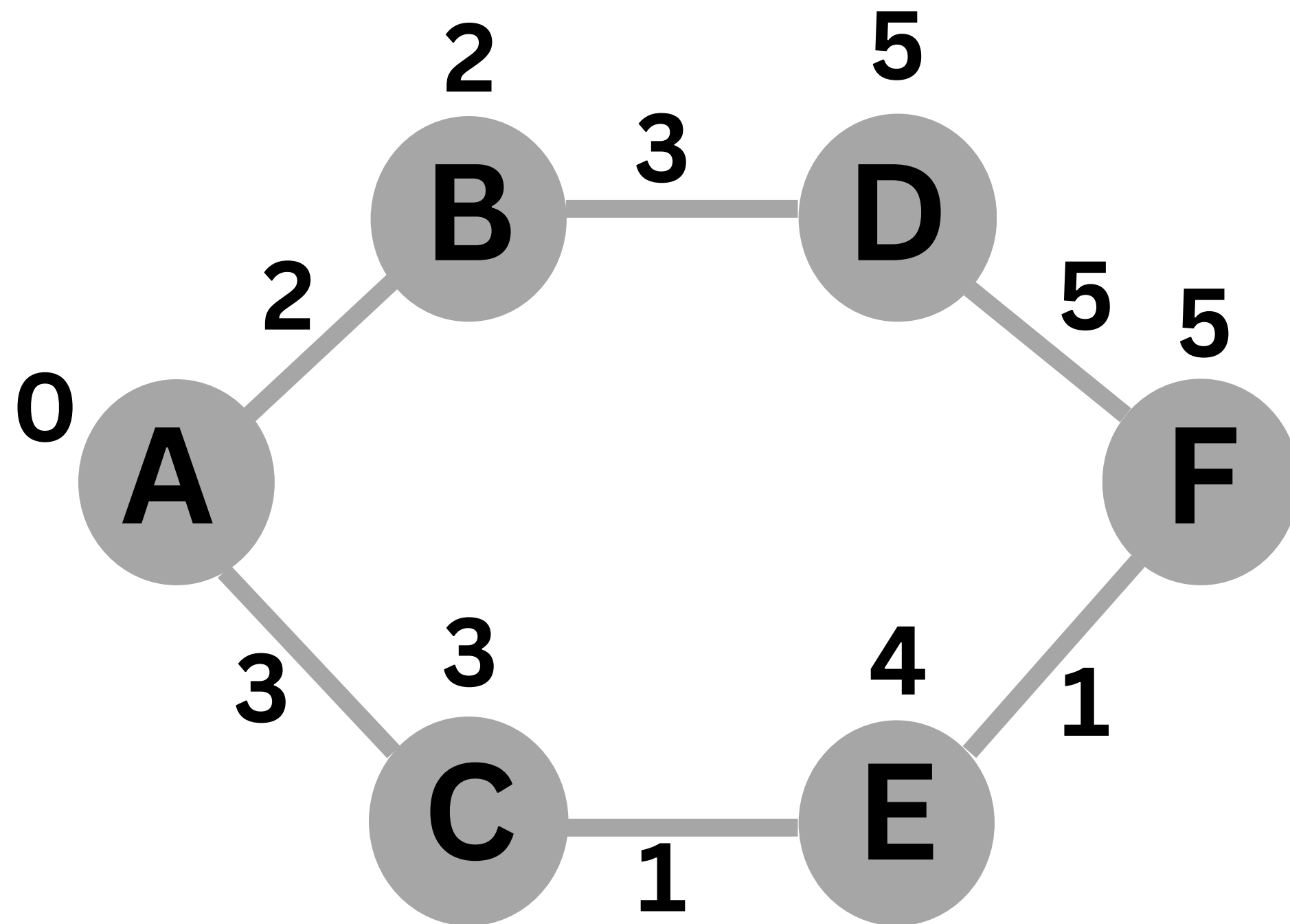
NODE	SHORTEST DISTANCE	PREVIOUS NODE
A	0	
B	2	A
C	3	A
D	5	B
E	4	C
F	5	E

DIJKSTRA'S ALGORITHM



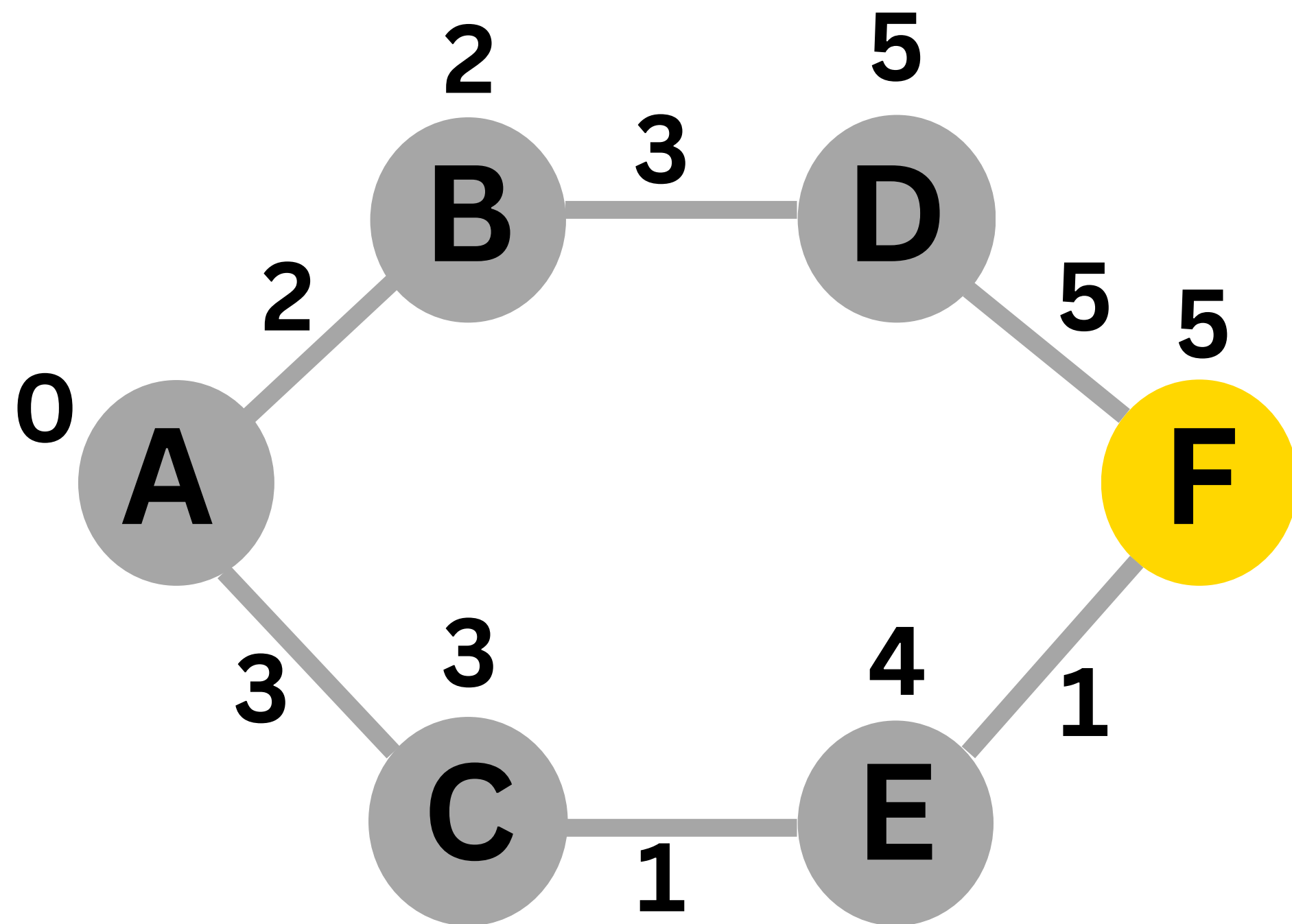
NODE	SHORTEST DISTANCE	PREVIOUS NODE
A	0	
B	2	A
C	3	A
D	5	B
E	4	C
F	5	E

DIJKSTRA'S ALGORITHM



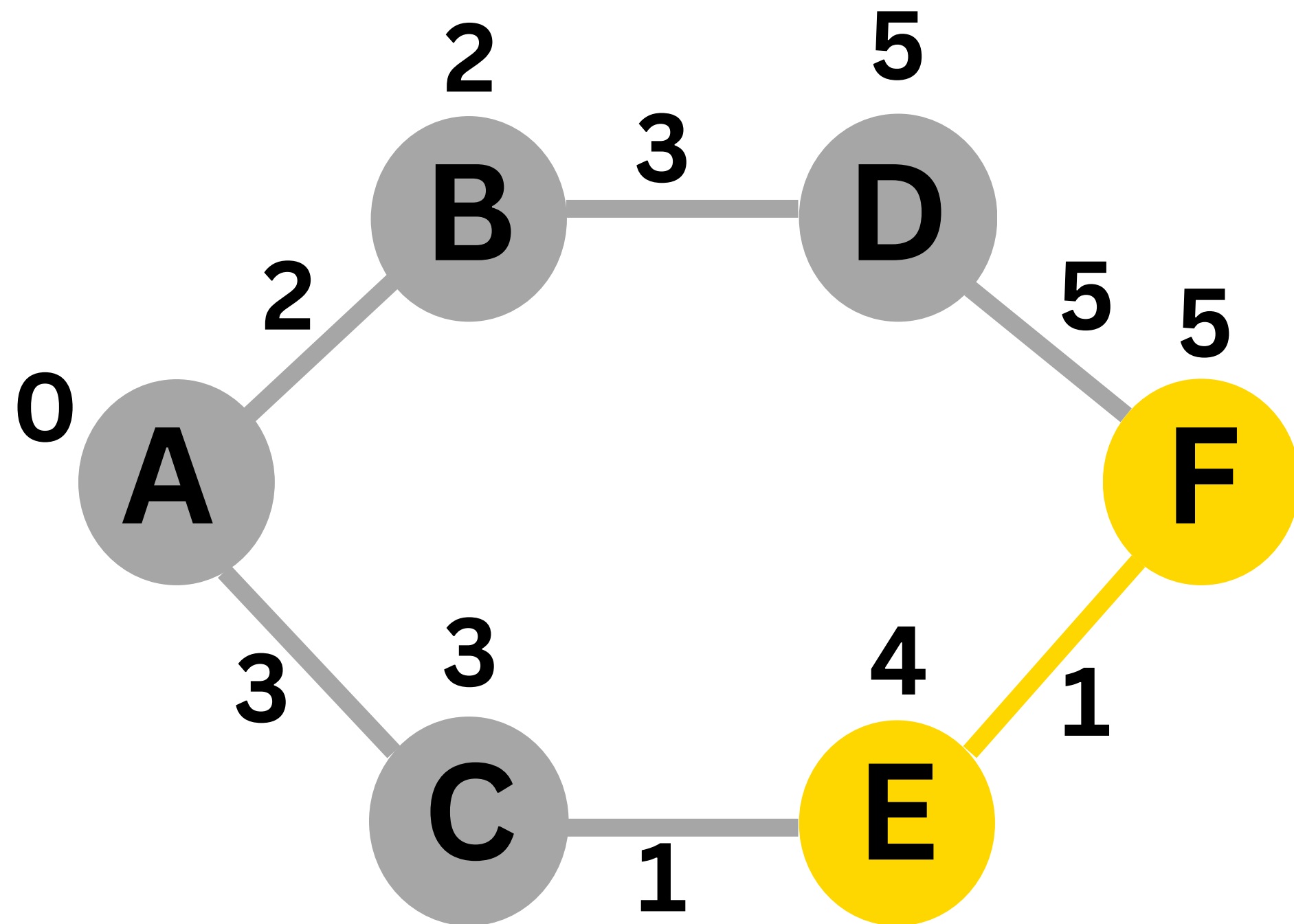
NODE	SHORTEST DISTANCE	PREVIOUS NODE
A	0	
B	2	A
C	3	A
D	5	B
E	4	C
F	5	E

DIJKSTRA'S ALGORITHM



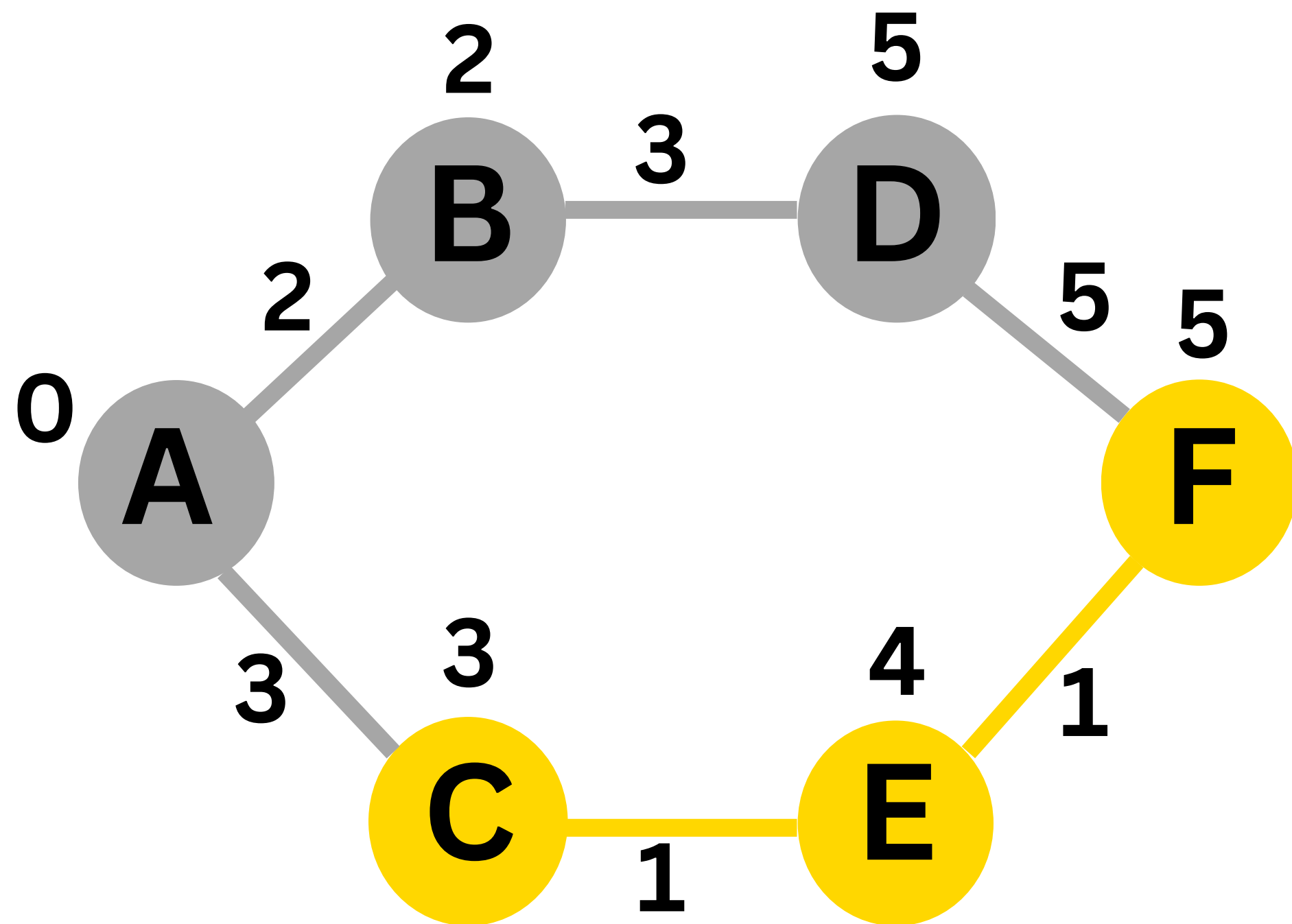
NODE	SHORTEST DISTANCE	PREVIOUS NODE
A	0	
B	2	A
C	3	A
D	5	B
E	4	C
F	5	E

DIJKSTRA'S ALGORITHM



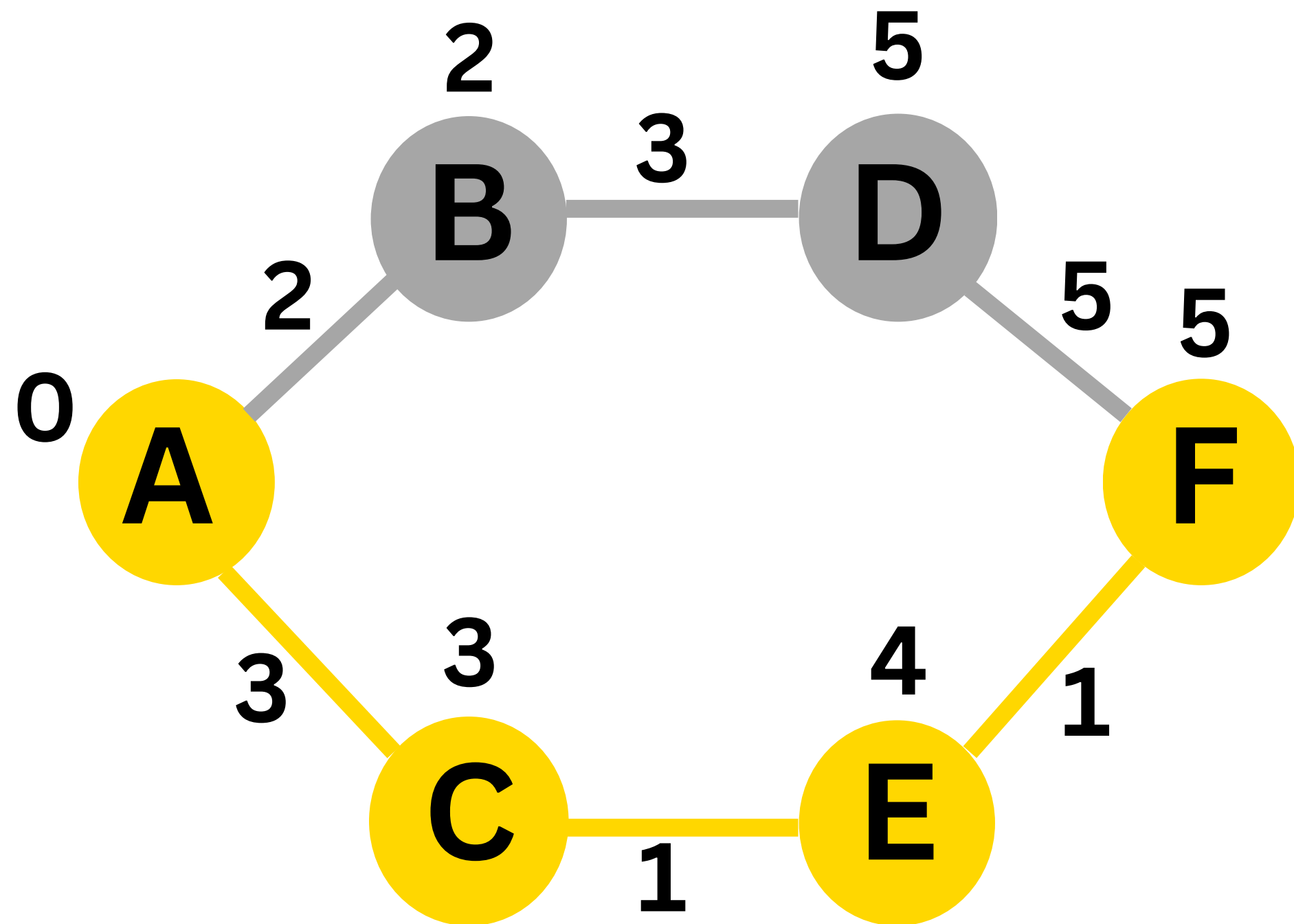
NODE	SHORTEST DISTANCE	PREVIOUS NODE
A	0	
B	2	A
C	3	A
D	5	B
E	4	C
F	5	E

DIJKSTRA'S ALGORITHM



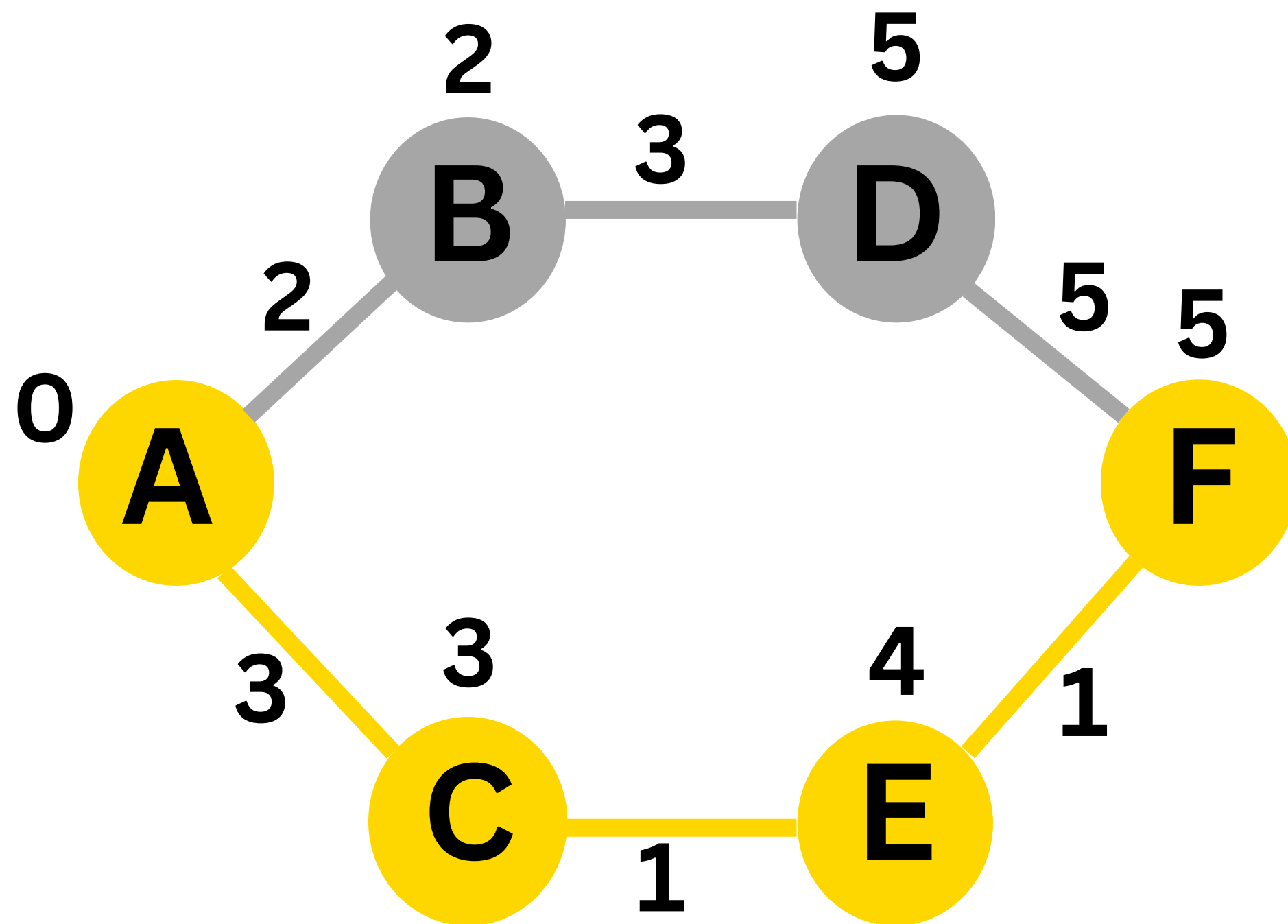
NODE	SHORTEST DISTANCE	PREVIOUS NODE
A	0	
B	2	A
C	3	A
D	5	B
E	4	C
F	5	E

DIJKSTRA'S ALGORITHM



NODE	SHORTEST DISTANCE	PREVIOUS NODE
A	0	
B	2	A
C	3	A
D	5	B
E	4	C
F	5	E

DIJKSTRA'S ALGORITHM



NODE	SHORTEST DISTANCE	PREVIOUS NODE
A	0	
B	2	A
C	3	A
D	5	B
E	4	C
F	5	E

TaDa! we got the shortest path.