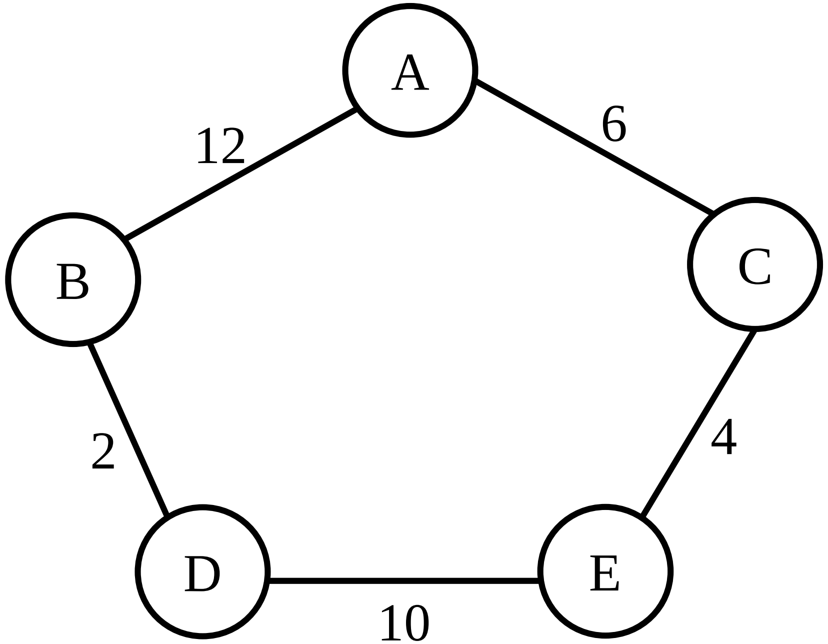
Shortest Path Problems

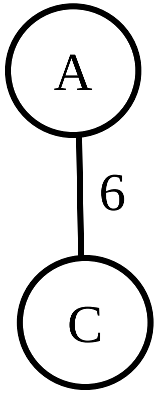
- Find the shortest path from V_1 to V_2 .
- Find the shortest path from V_1 to every vertex (Single Source Shortest Paths).
- Find the shortest path for all pairs of vertices (All Sources Shortest Paths).

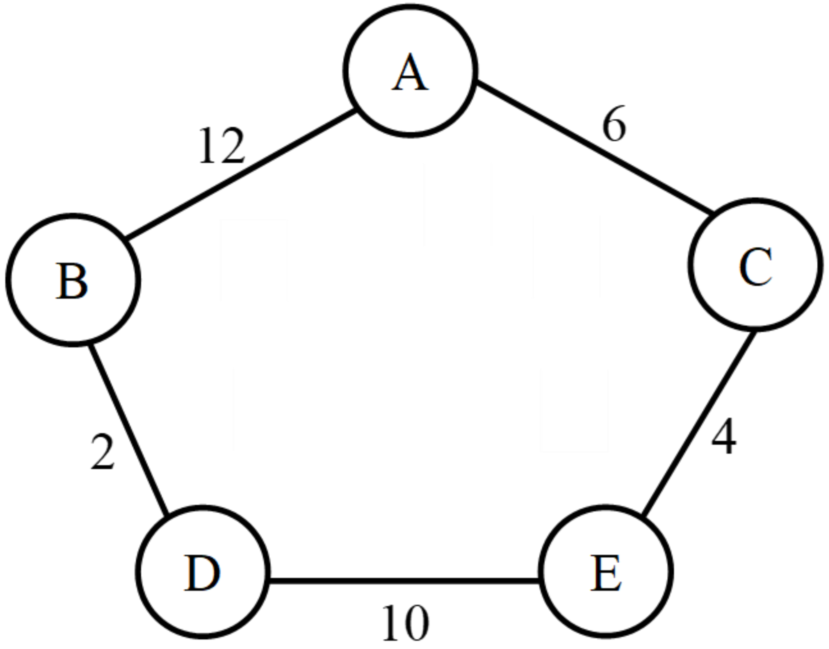
Single Source Shortest Paths (Dijkstra’s Algorithm)



Settled/Known: A

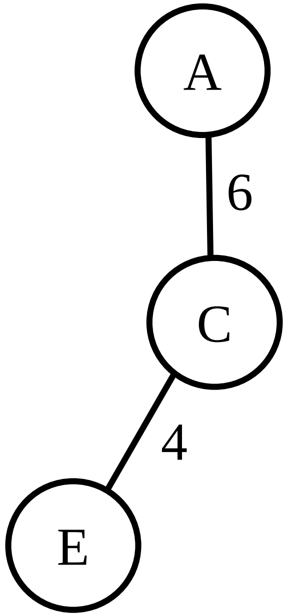
Candidates: B [distance (path cost) = 12]
C [distance (path cost) = 6]

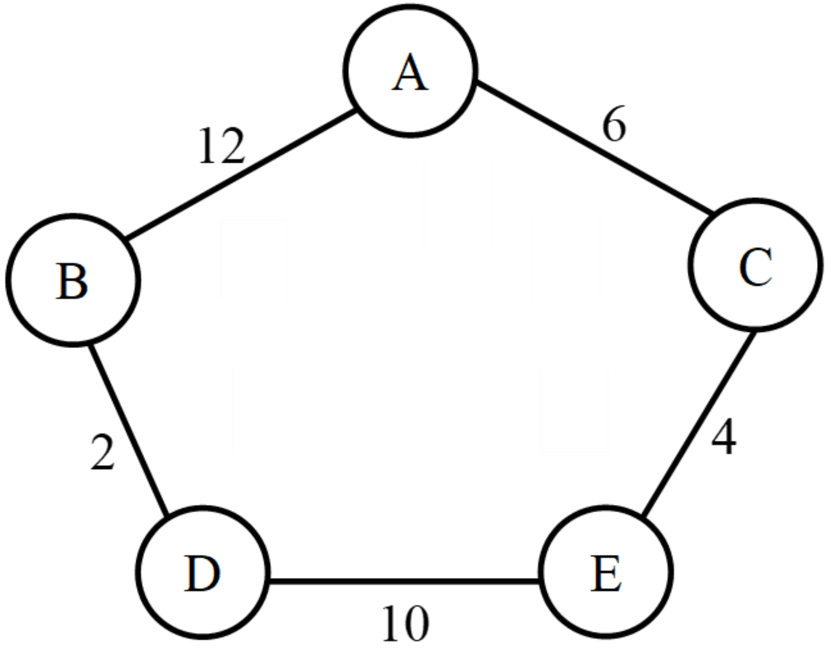




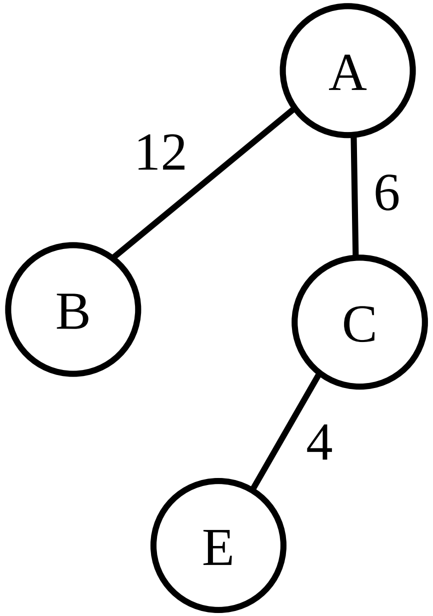
Settled/Known: A, C

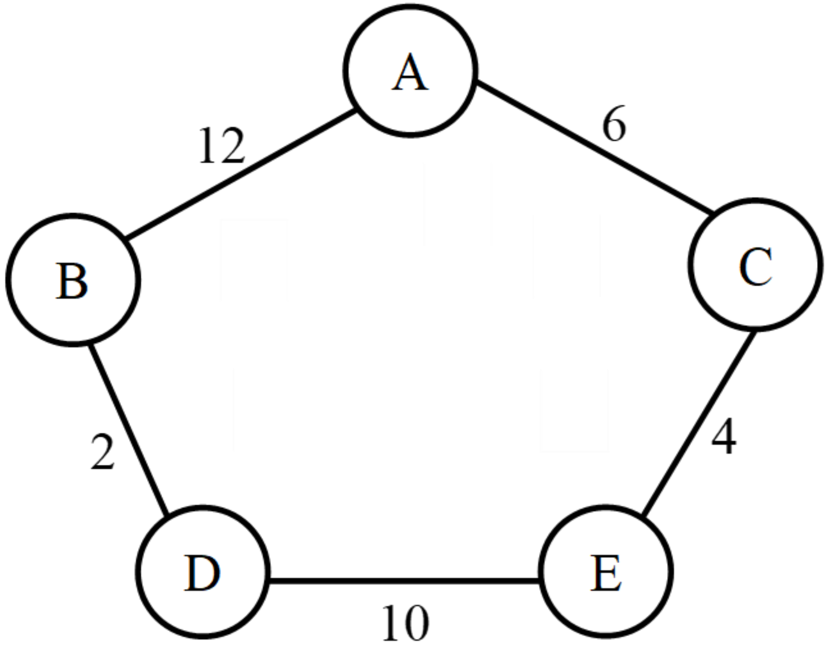
Candidates: B [distance (path cost) = 12]
E [distance (path cost) = 10]



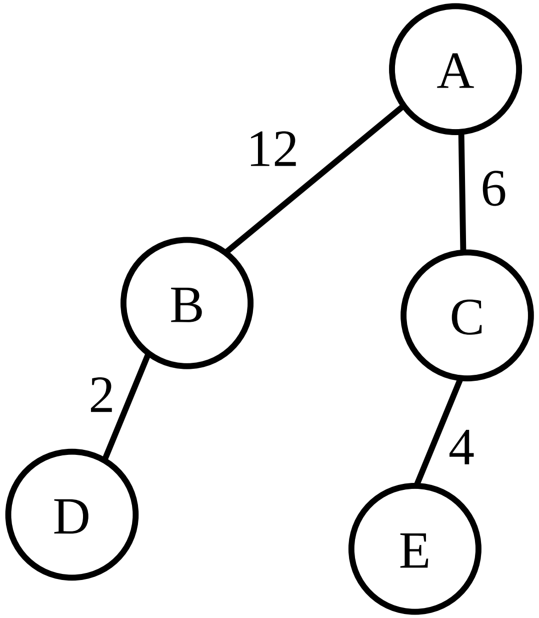


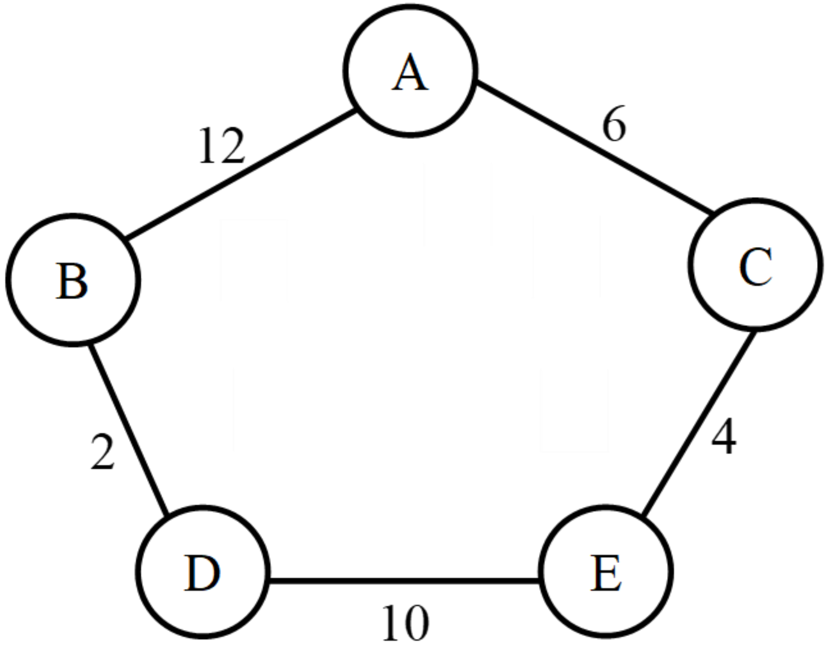
Settled/Known: A, C, E
Candidates: B [distance (path cost) = 12]
D [distance (path cost) = 20]





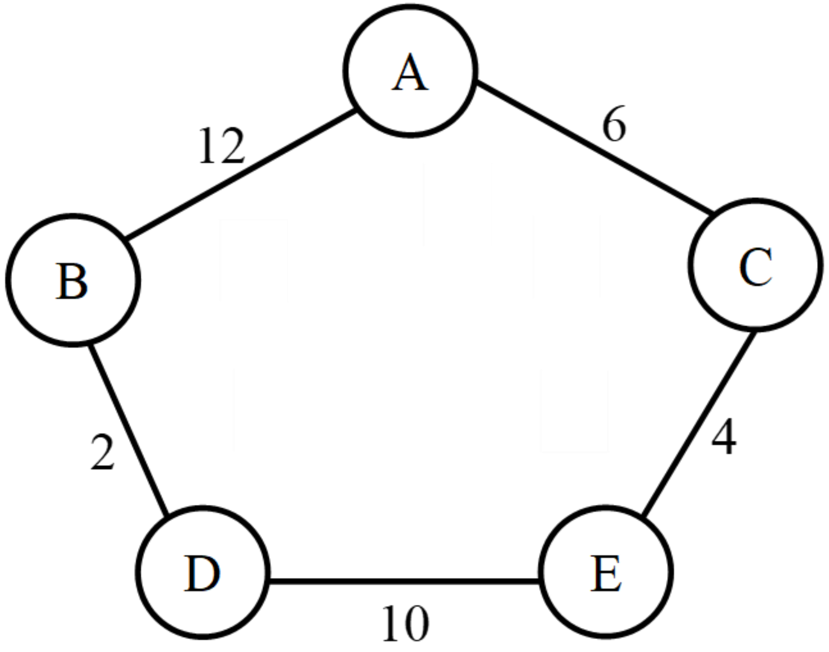
Settled/Known: A, B, C, E
Candidates: D [distance (path cost) = 14]





Implementation

Distance table	0	12	6	∞	∞
Index	A	B	C	D	E



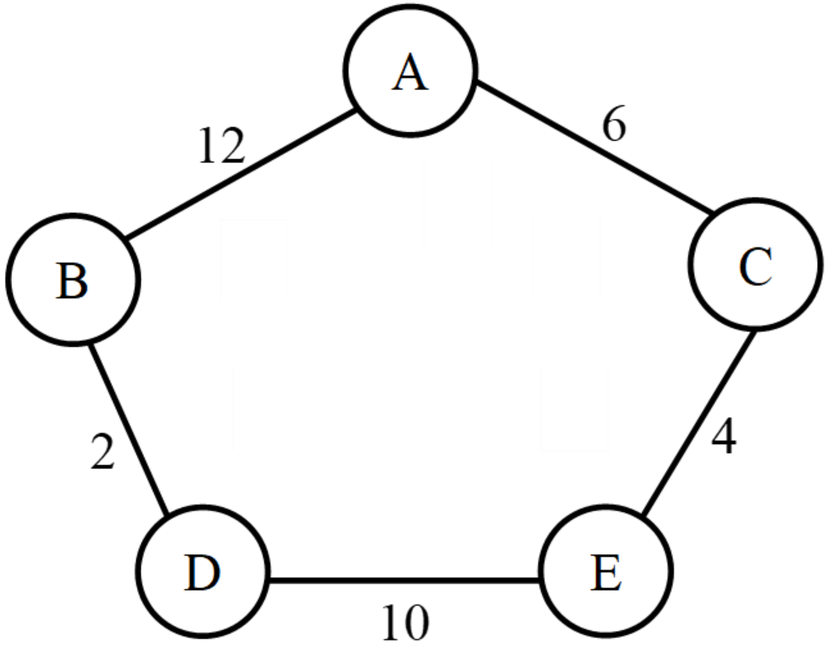
Implementation

Pick the smallest distance (consider only the unvisited vertices) and update the distance table:

Distance table	0	12	6	∞	∞

Index	A	B	C	D	E

Distance table	0	12	6	∞	10



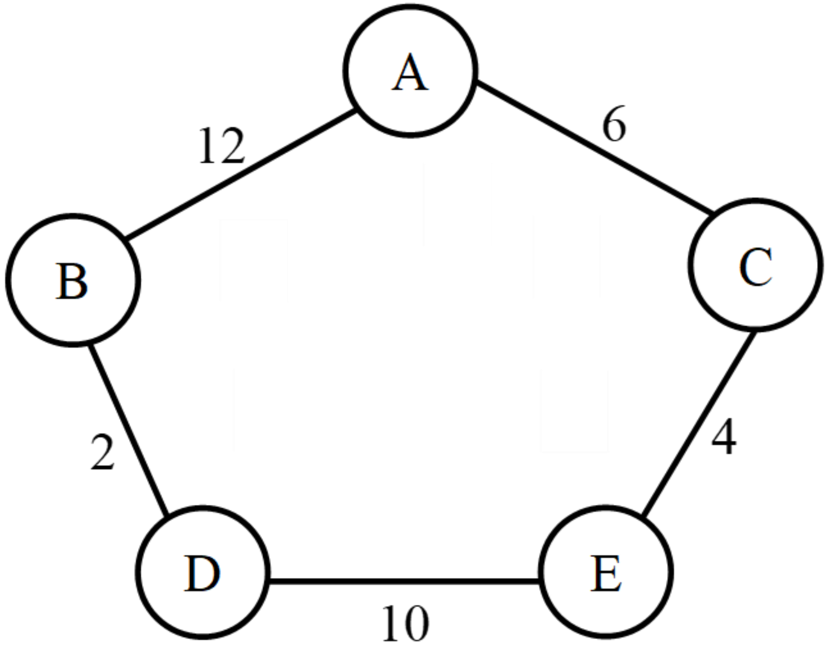
Implementation

Pick the smallest distance (consider only the unvisited vertices) and update the distance table:

Distance table	0	12	6	∞	10

Index	A	B	C	D	E

Distance table	0	12	6	20	10



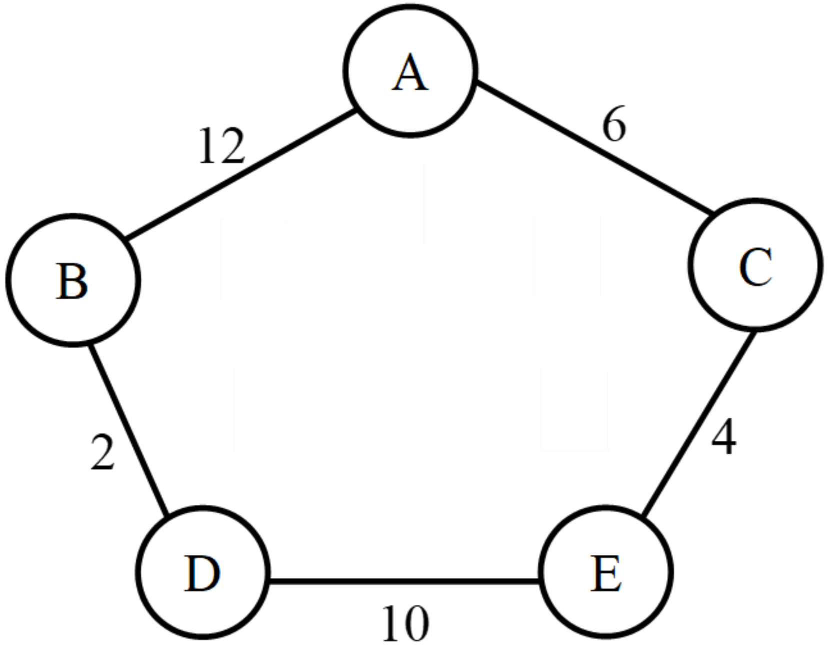
Implementation

Pick the smallest distance (consider only the unvisited vertices) and update the distance table:

Distance table	0	12	6	20	10

Index	A	B	C	D	E

Distance table	0	12	6	14	10


A
12
6
B
C
2
4
D
E
10