Task 2: Explanation

We start off from the start node, and maintain a table as shown in the steps below, which records the information like predecessor node, the distance to the nodes and whether a node has been visited or not. At every step, we choose to visit that node from the table which is not yet visited, **and** has the least distance value. Then we check from the graph all the nodes that are accessible from this node that we have just visited, and update those nodes if the distance to reach them from the new node is less than than they previous distance which was transcribed in the table. We continue this procedure until we have visited the end node.

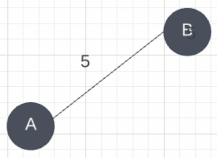
STEP 1: visit A

A:	A

Node	Distance	Previous Node	Visited
A	0	-	True
B	5	A	False
H	8	A	False
E	9	A	False
C	NaN	-	False
F	NaN	-	False
D	NaN	-	False
G	NaN	-	False

STEP 2: Visit B

A:	A
B:	A -> B



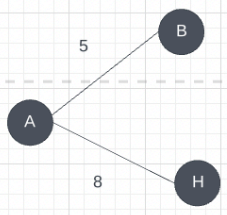
Node	Distance	Previous Node	Visited
A	0	-	True
B	5	A	True
H	8	A	False
E	9	A	False
C	17	B	False
F	NaN	-	False
D	20	B	False
G	NaN	-	False

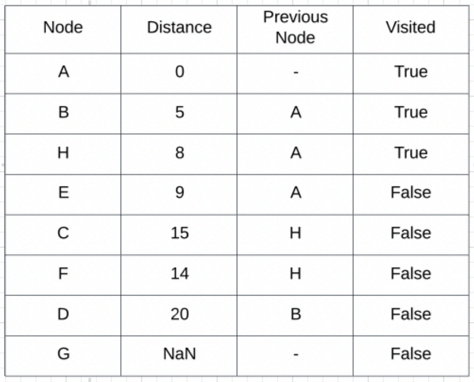
STEP 3: Visit H

A:	A
B:	A -> B
H:	A -> H

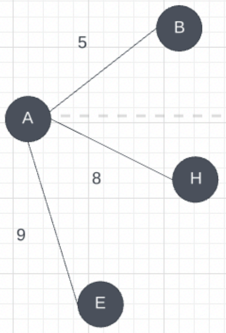
STEP 4: Visit E

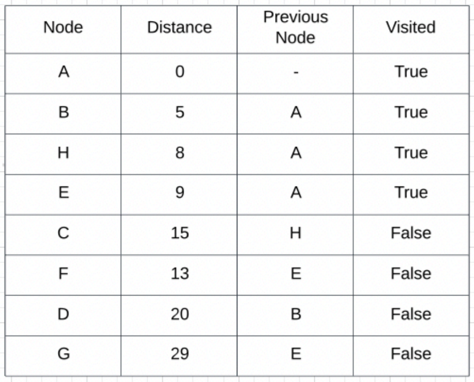
A:	A
B:	A -> B
H:	A -> H
E:	A -> E





Node	Distance	Previous Node	Visited
A	0	-	True
B	5	A	True
H	8	A	True
E	9	A	False
C	15	H	False
F	14	H	False
D	20	B	False
G	NaN	-	False





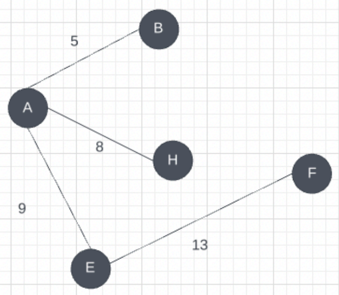
Node	Distance	Previous Node	Visited
A	0	-	True
B	5	A	True
H	8	A	True
E	9	A	True
C	15	H	False
F	13	E	False
D	20	B	False
G	29	E	False

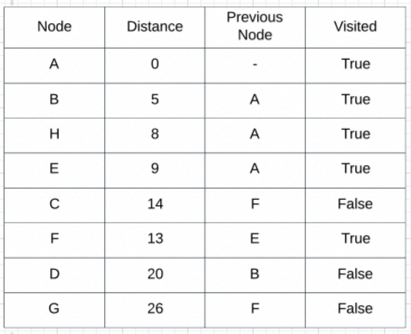
STEP 5: Visit F

A:	A
B:	A -> B
H:	A -> H
E:	A -> E
F:	A -> E -> F

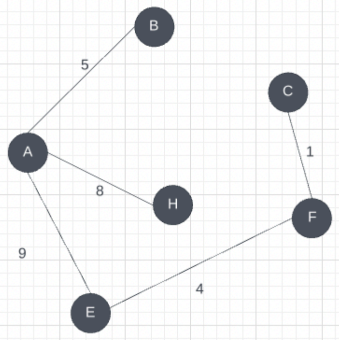
STEP 6: Visit C

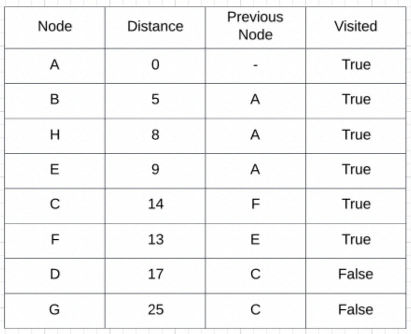
A:	A
B:	A -> B
H:	A -> H
E:	A -> E
F:	A -> E -> F
C:	A -> E -> F -> C





Node	Distance	Previous Node	Visited
A	0	-	True
B	5	A	True
H	8	A	True
E	9	A	True
C	14	F	False
F	13	E	True
D	20	B	False
G	26	F	False



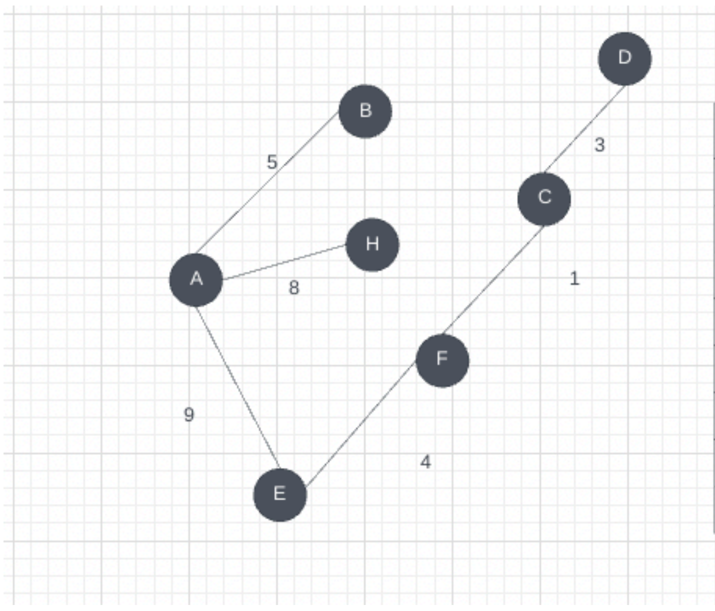


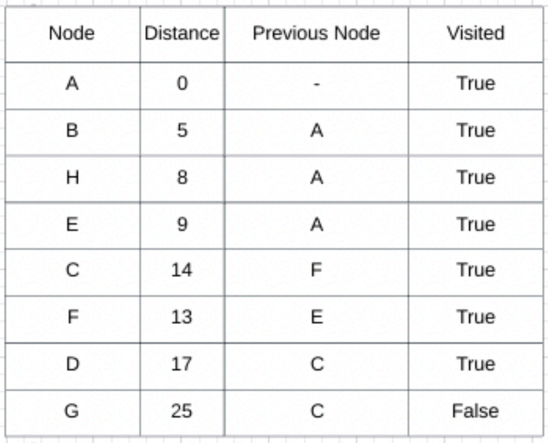
Node	Distance	Previous Node	Visited
A	0	-	True
B	5	A	True
H	8	A	True
E	9	A	True
C	14	F	True
F	13	E	True
D	17	C	False
G	25	C	False

STEP 7: Visit D

B:	A -> B
H:	A -> H
E:	A -> E
F:	A -> E -> F

C:	A -> E -> F -> C
D:	A -> E -> F -> C -> D

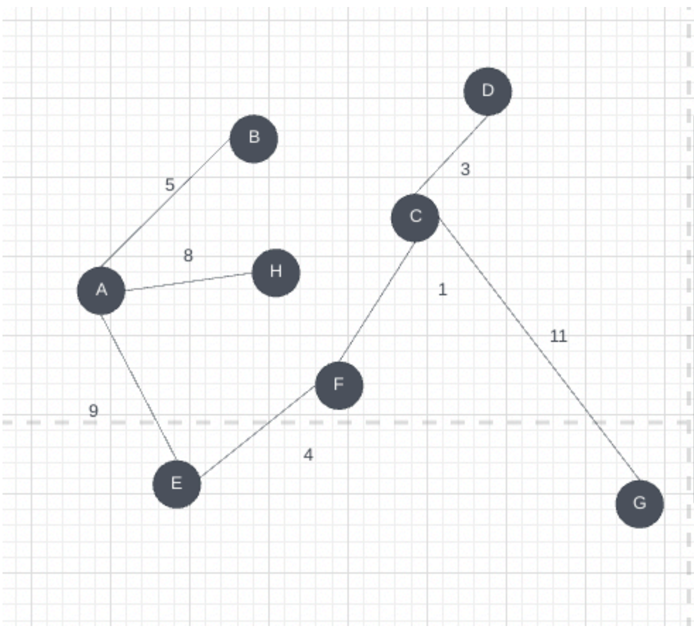


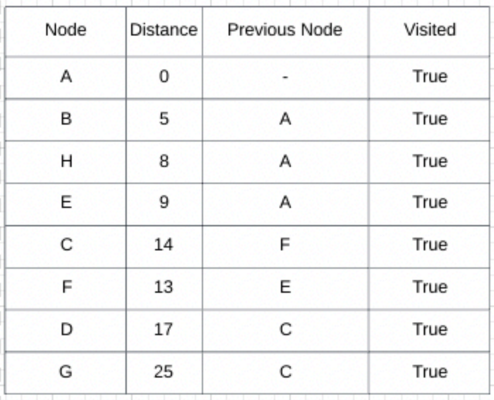


Node	Distance	Previous Node	Visited
A	0	-	True
B	5	A	True
H	8	A	True
E	9	A	True
C	14	F	True
F	13	E	True
D	17	C	True
G	25	C	False

STEP 8: Visit G

B:	A -> B
H:	A -> H
E:	A -> E
F:	A -> E -> F
C:	A -> E -> F -> C
D:	A -> E -> F -> C -> D
G:	A -> E -> F -> C -> G





Node	Distance	Previous Node	Visited
A	0	-	True
B	5	A	True
H	8	A	True
E	9	A	True
C	14	F	True
F	13	E	True
D	17	C	True
G	25	C	True