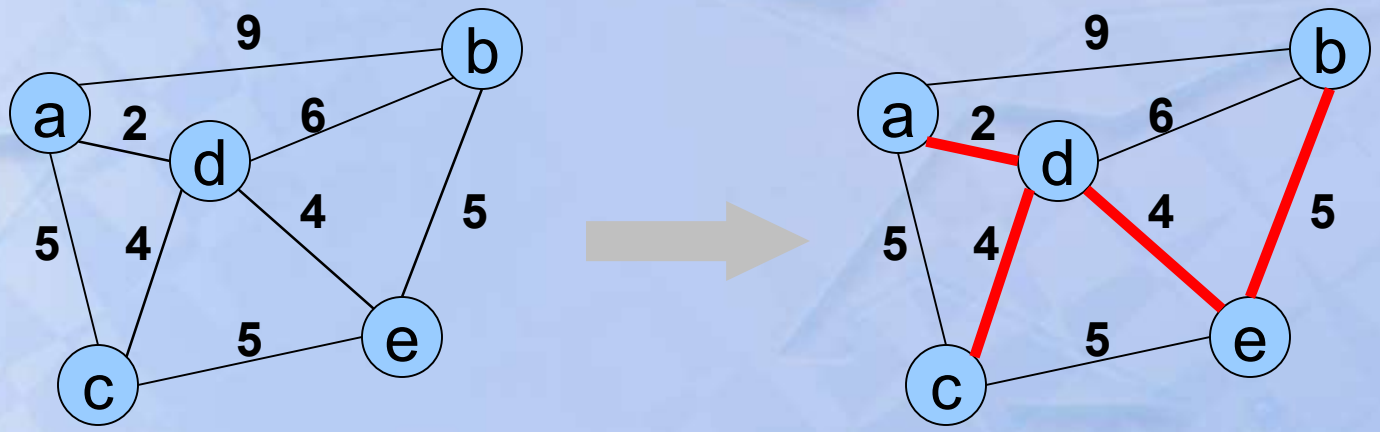
Minimum Spanning Tree

Minimum Spanning Trees (MST)

A **minimum spanning tree** is a subgraph of an undirected weighted graph ***G***, such that

- it is a tree (i.e., it is acyclic)
- it covers ***(spans)*** all the vertices ***V*** contains ***|V| - 1*** edges
- the total cost is the minimum among all possible spanning trees
- not necessarily unique


9
b
a
2
d
6
5
4
4
5
5
e
c
9
b
a
2
d
6
5
4
4
5
5
e
c

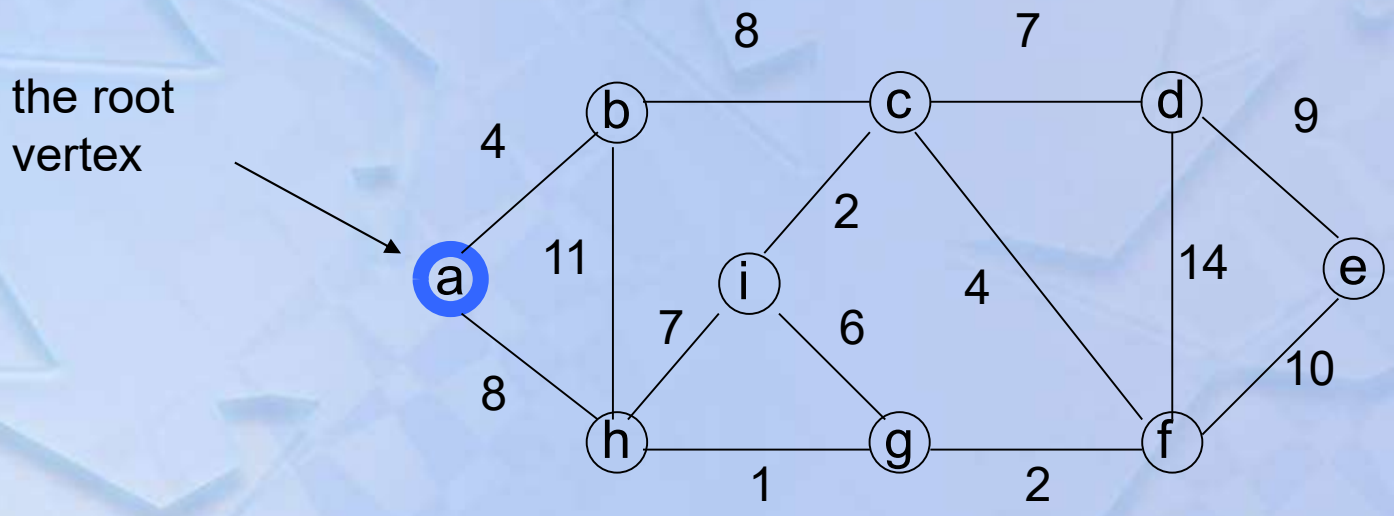
Algorithm for MST

- Kruskal algorithm
  - A MST can be grown from a forest of spanning trees by adding the smallest edge connecting two spanning trees.
- Prims Algorithm
  - A MST can be grown from the current spanning tree by adding the nearest vertex and the edge connecting the nearest vertex to the MST.

Prim's Algorithm

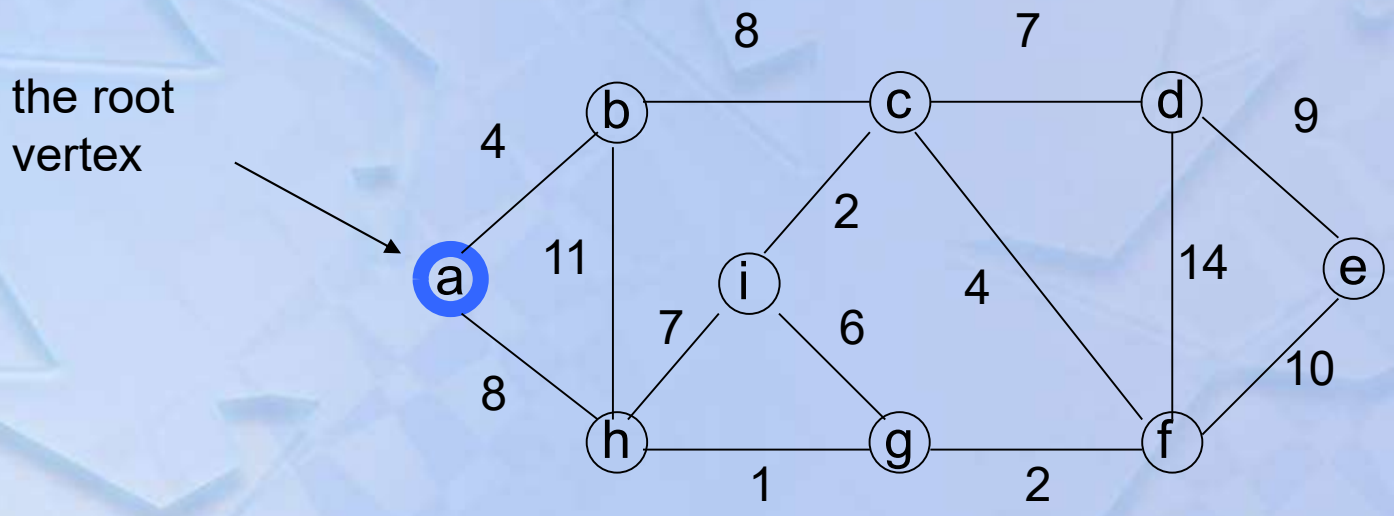
- Prim's algorithm starts from an vertex called root vertex.
- At each step it finds an minimum edge until it spans all the vertex in V.

The execution of Prim's algorithm



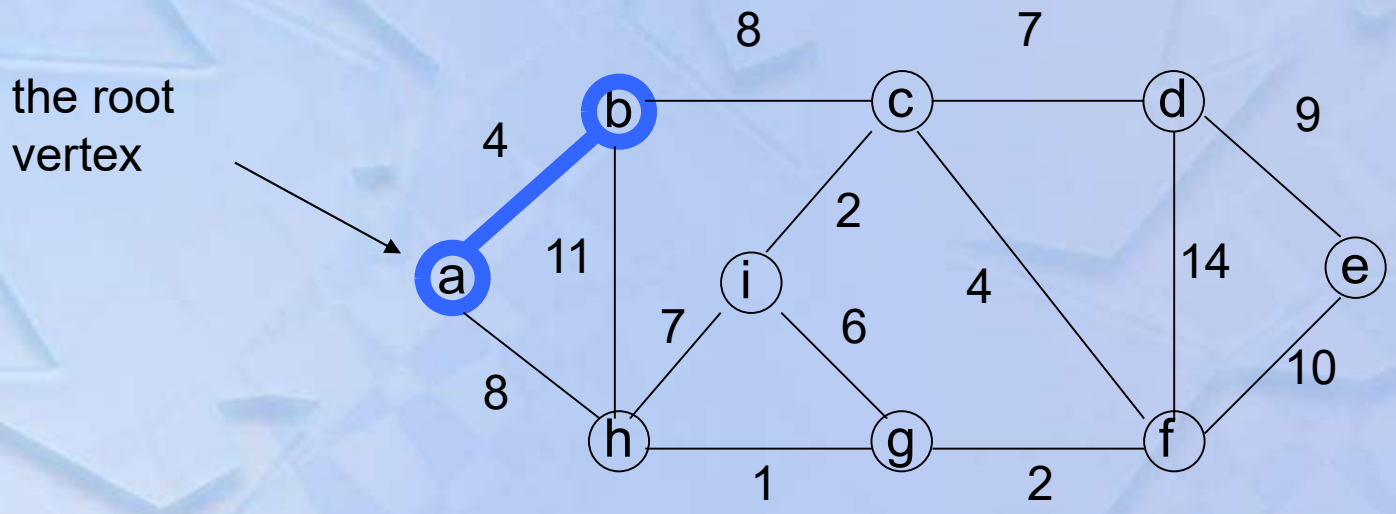
V	a	b	c	d	e	f	g	h	i
T	1	0	0	0	0	0	0	0	0
Key	0	-	-	-	-	-	-	-	-
π	-1	-	-	-	-	-	-	-	-

The execution of Prim's algorithm(moderate part)



V	a	b	c	d	e	f	g	h	i
T	1	**0**	0	0	0	0	0	0	0
Key	0	**4**	-	-	-	-	-	8	-
π	-1	**a**	-	-	-	-	-	a	-

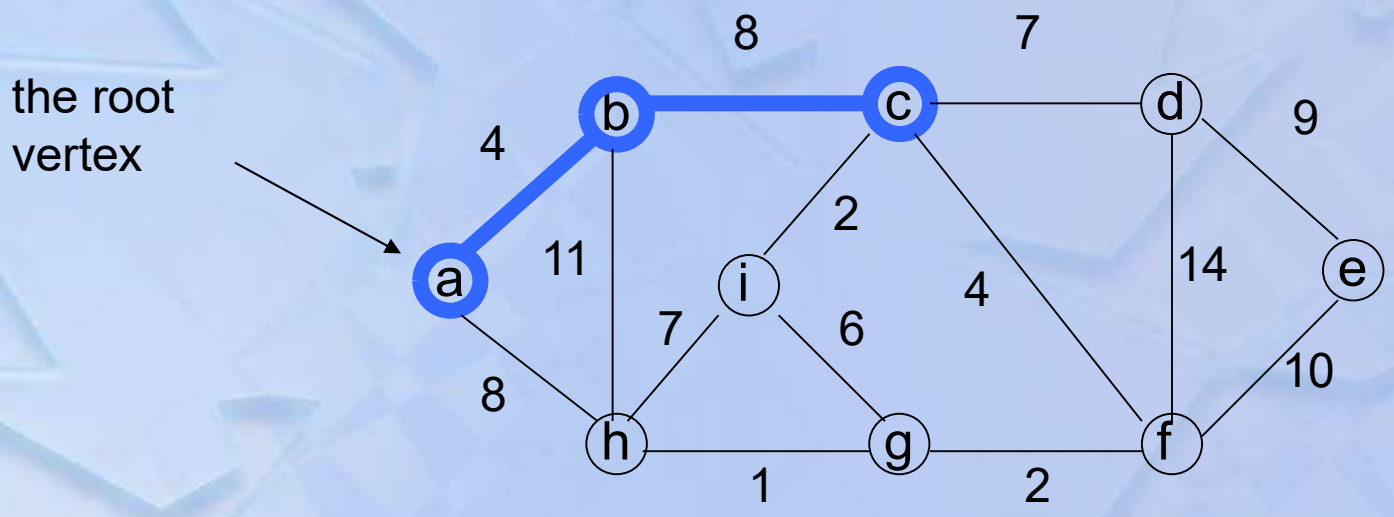
The execution of Prim's algorithm(moderate part)



Important: Update Key[v] only if T[v]==0

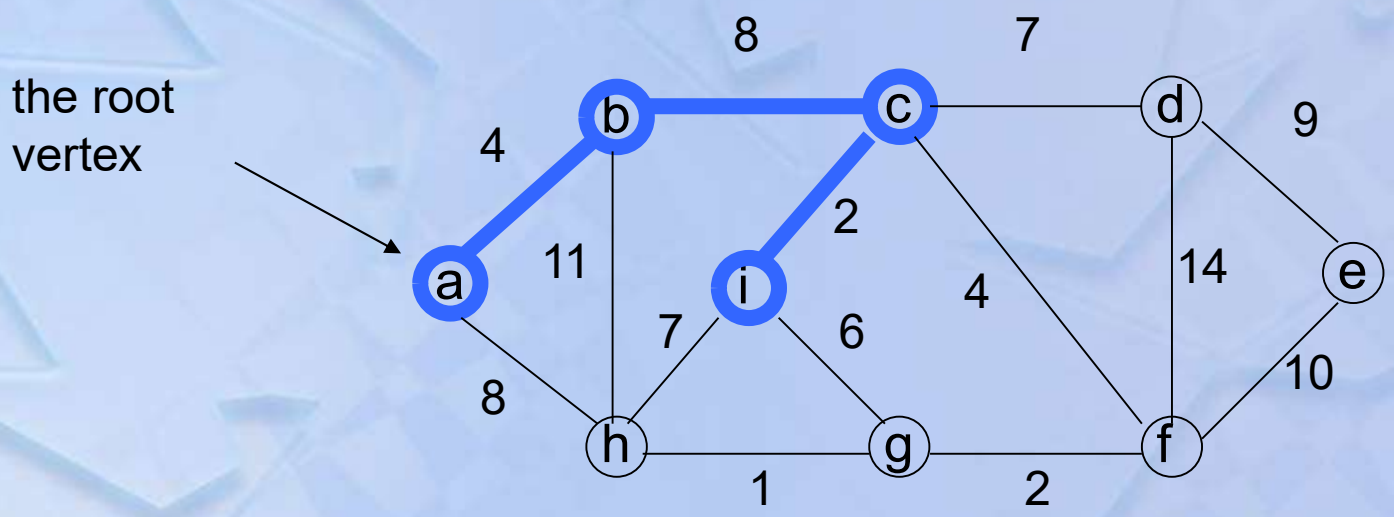
V	a	b	c	d	e	f	g	h	i
T	1	1	**0**	0	0	0	0	0	0
Key	0	4	**8**	-	-	-	-	8	-
π	-1	a	**b**	-	-	-	-	a	-

The execution of Prim's algorithm(moderate part)



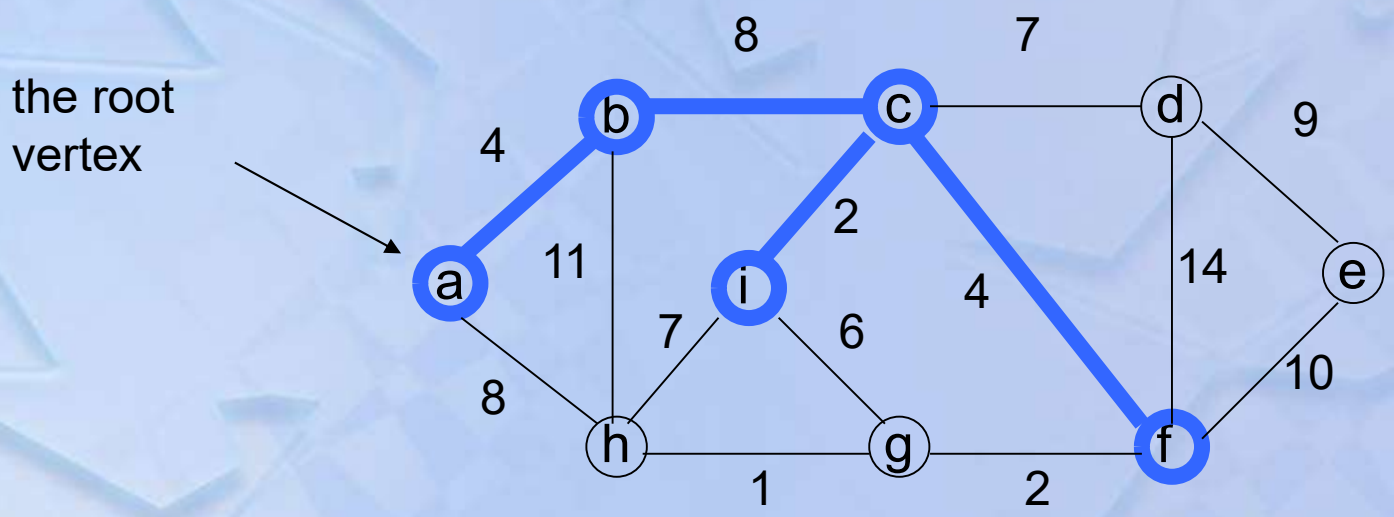
V	a	b	c	d	e	f	g	h	i
T	1	1	1	0	0	0	0	0	**0**
Key	0	4	8	7	-	4	-	8	**2**
π	-1	a	b	c	-	c	-	a	**c**

The execution of Prim's algorithm(moderate part)



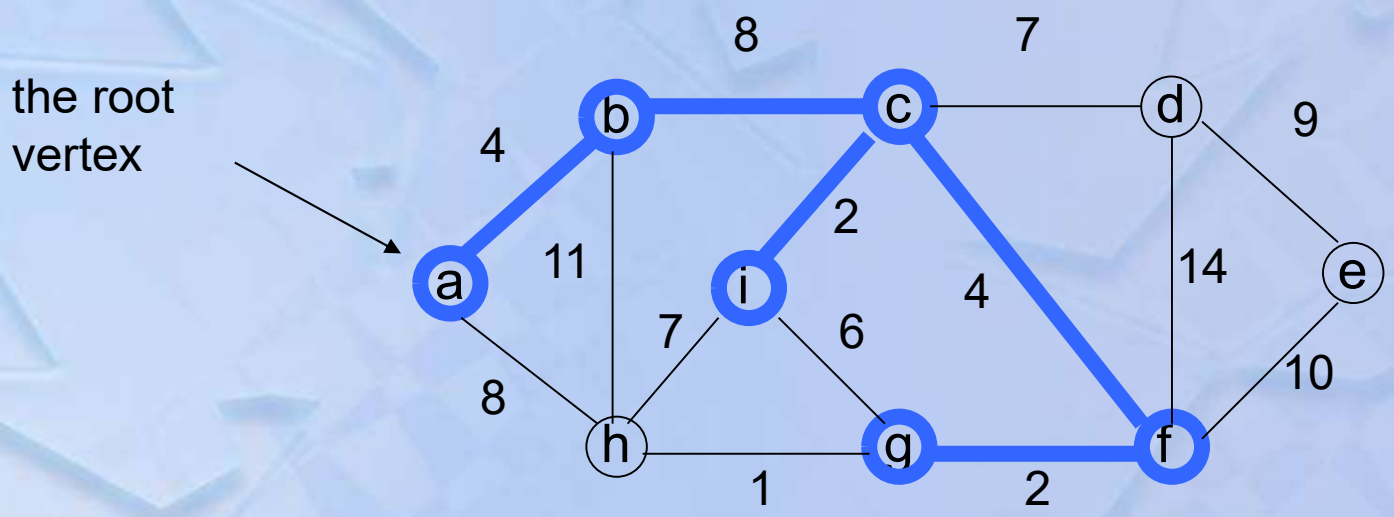
V	a	b	c	d	e	f	g	h	i
T	1	1	1	0	0	**0**	0	0	1
Key	0	4	8	7	-	**4**	6	7	2
π	-1	a	b	c	-	**d**	i	i	c

The execution of Prim's algorithm(moderate part)



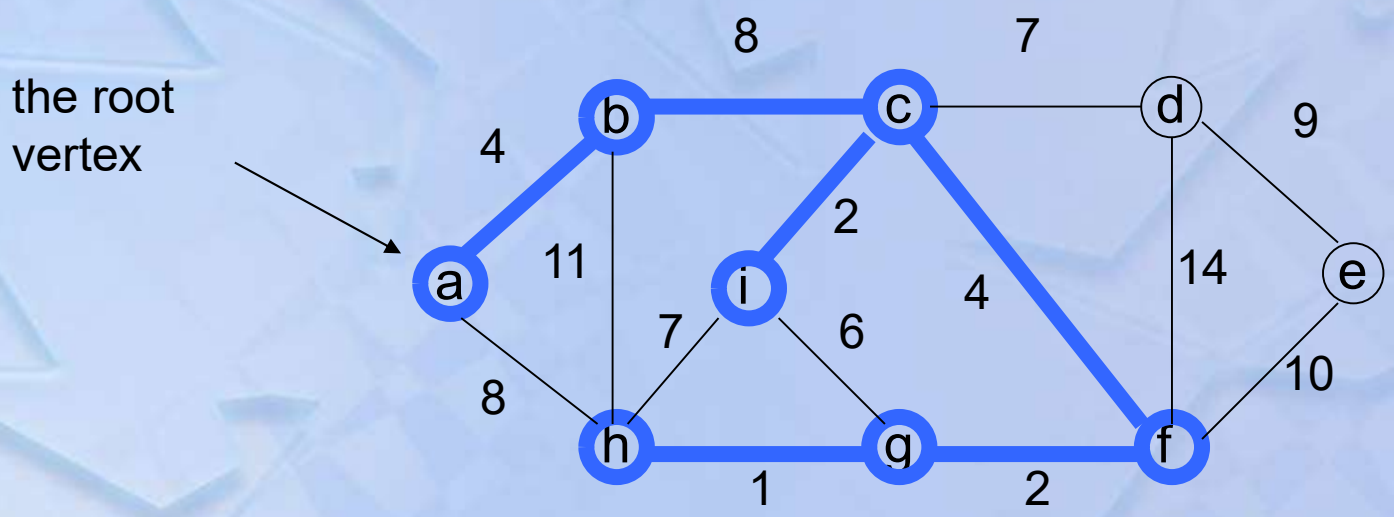
V	a	b	c	d	e	f	g	h	i
T	1	1	1	0	0	**1**	**0**	0	1
Key	0	4	8	7	10	**4**	**2**	7	2
π	-1	a	b	c	f	**c**	**f**	i	c

The execution of Prim's algorithm(moderate part)



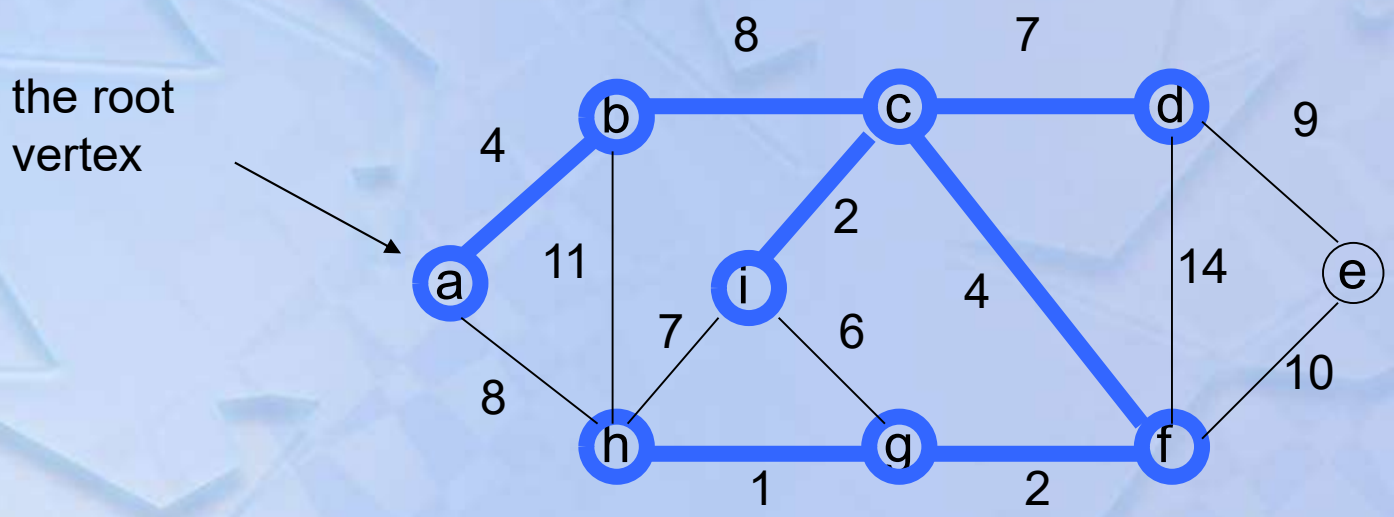
V	a	b	c	d	e	f	g	h	i
T	1	1	1	0	0	1	**1**	**0**	1
Key	0	4	8	7	10	4	**2**	**1**	2
π	-1	a	b	c	f	c	**f**	**g**	c

The execution of Prim's algorithm(moderate part)



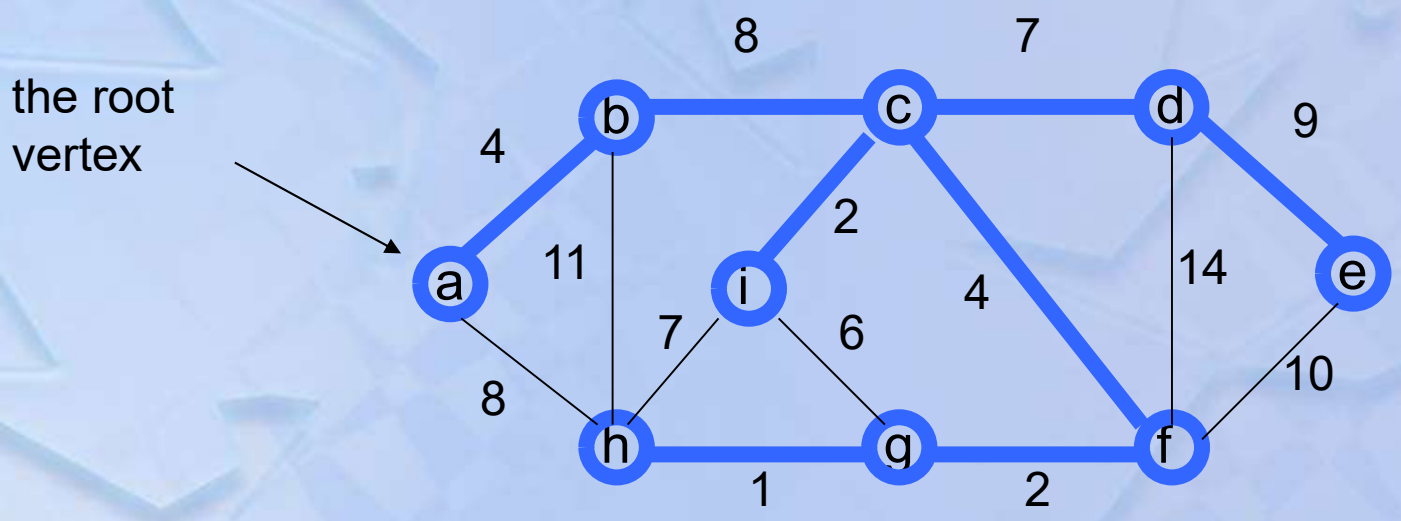
V	a	b	c	d	e	f	g	h	i
T	1	1	1	**0**	0	1	1	**1**	1
Key	0	4	8	**7**	10	4	2	**1**	2
π	-1	a	b	**c**	f	c	f	**g**	c

The execution of Prim's algorithm(moderate part)



V	a	b	c	d	e	f	g	h	i
T	1	1	1	**1**	**0**	1	1	1	1
Key	0	4	8	**7**	**9**	4	2	1	2
π	-1	a	b	**c**	**d**	c	f	g	c

The execution of Prim's algorithm(moderate part)

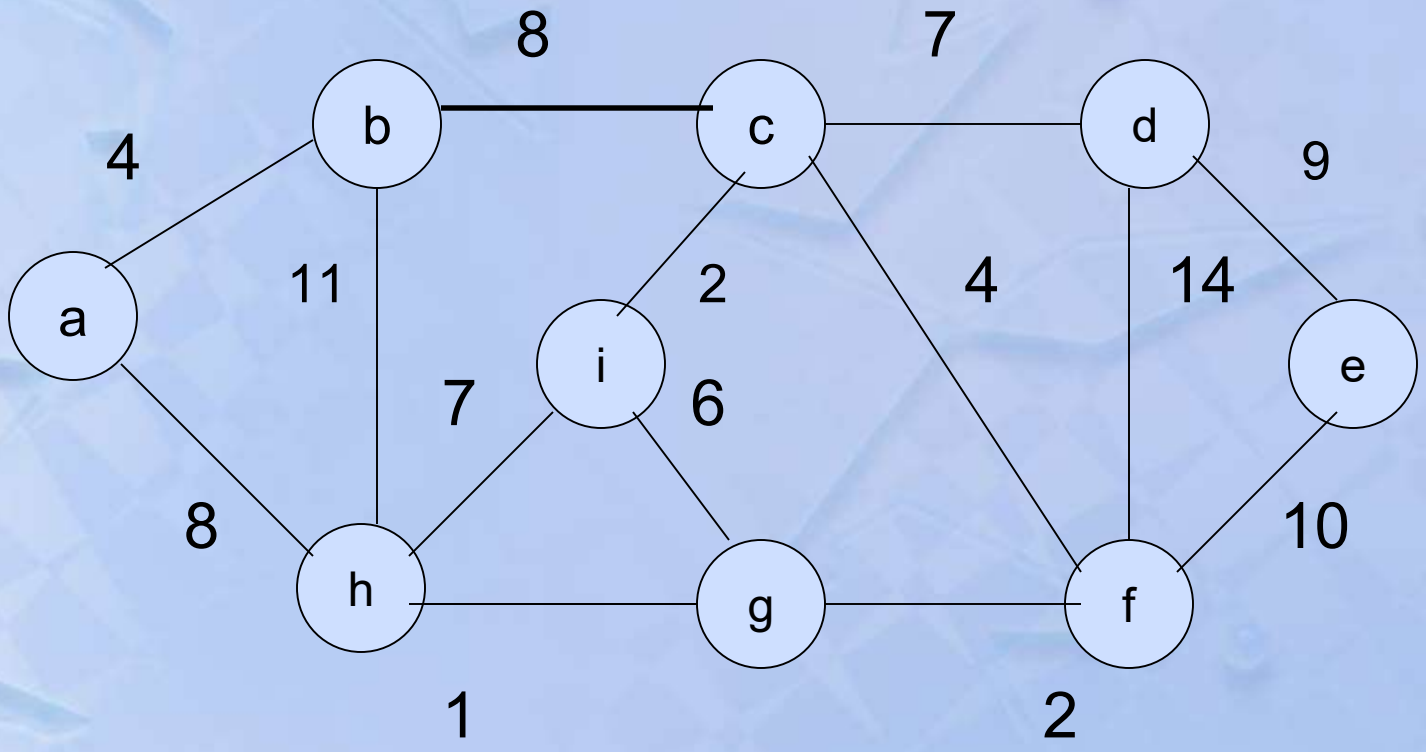


V	a	b	c	d	e	f	g	h	i
T	1	1	1	1	**1**	1	1	1	1
Key	0	4	8	7	**9**	4	2	1	2
π	-1	a	b	c	**d**	c	f	g	c

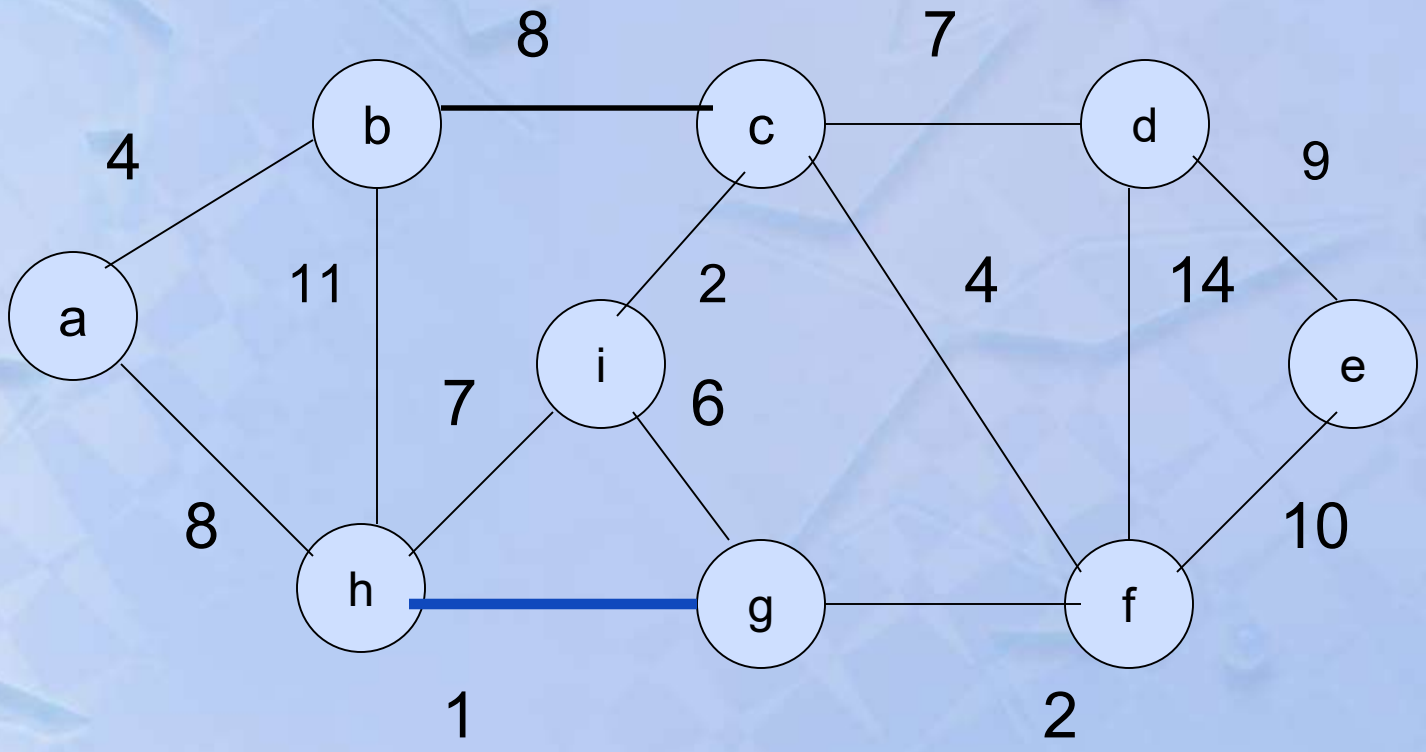
Kruskal's Algorithm

- In this case there is no root.
- At each step minimum edge is added without forming a cycle until it spans all the vertexes in the list.

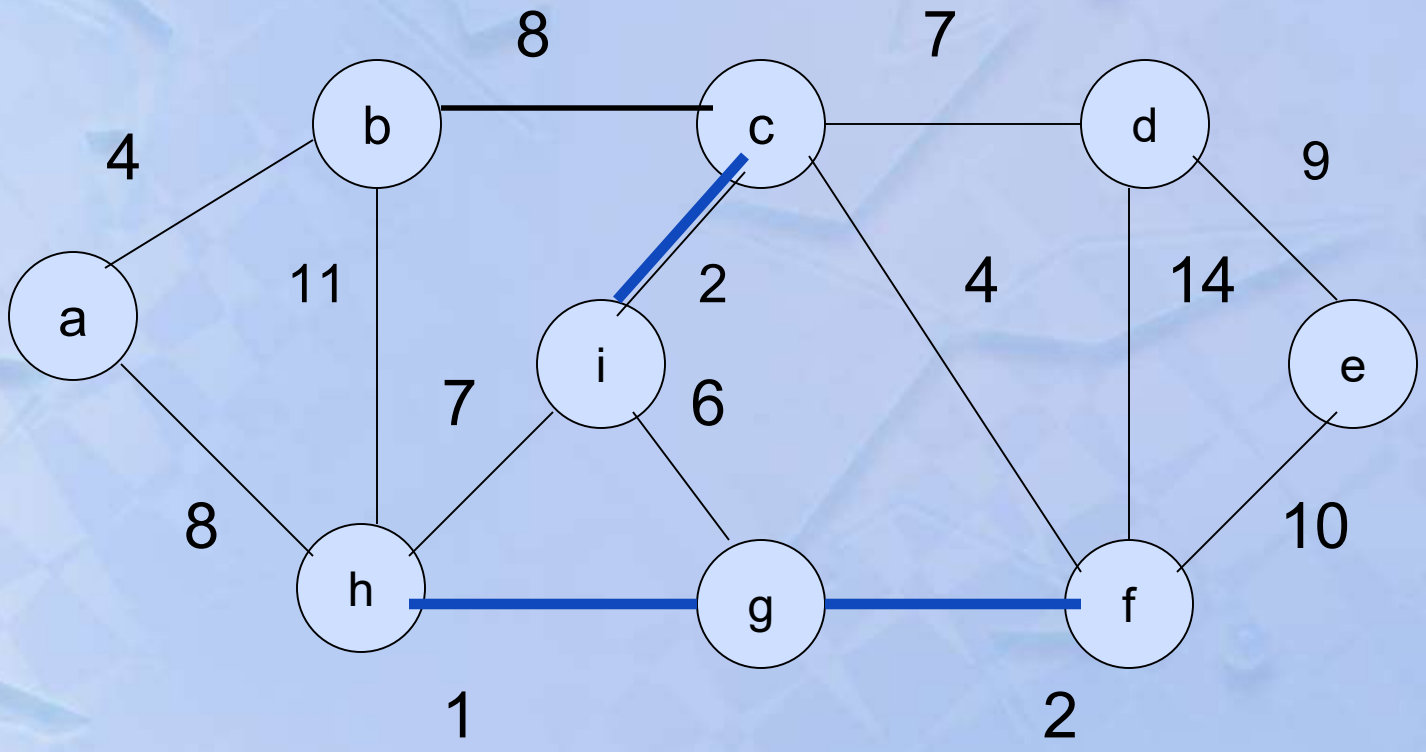
Initial Stage



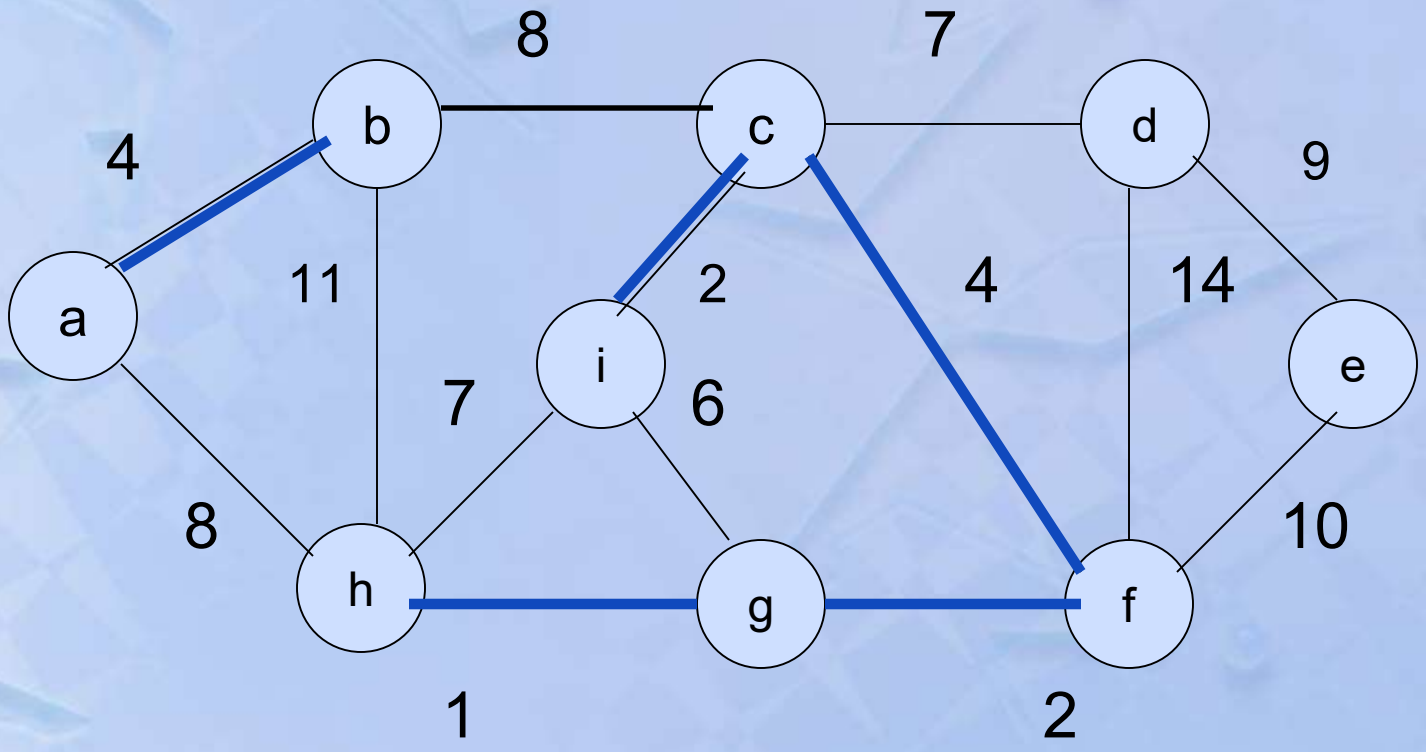
Step -1



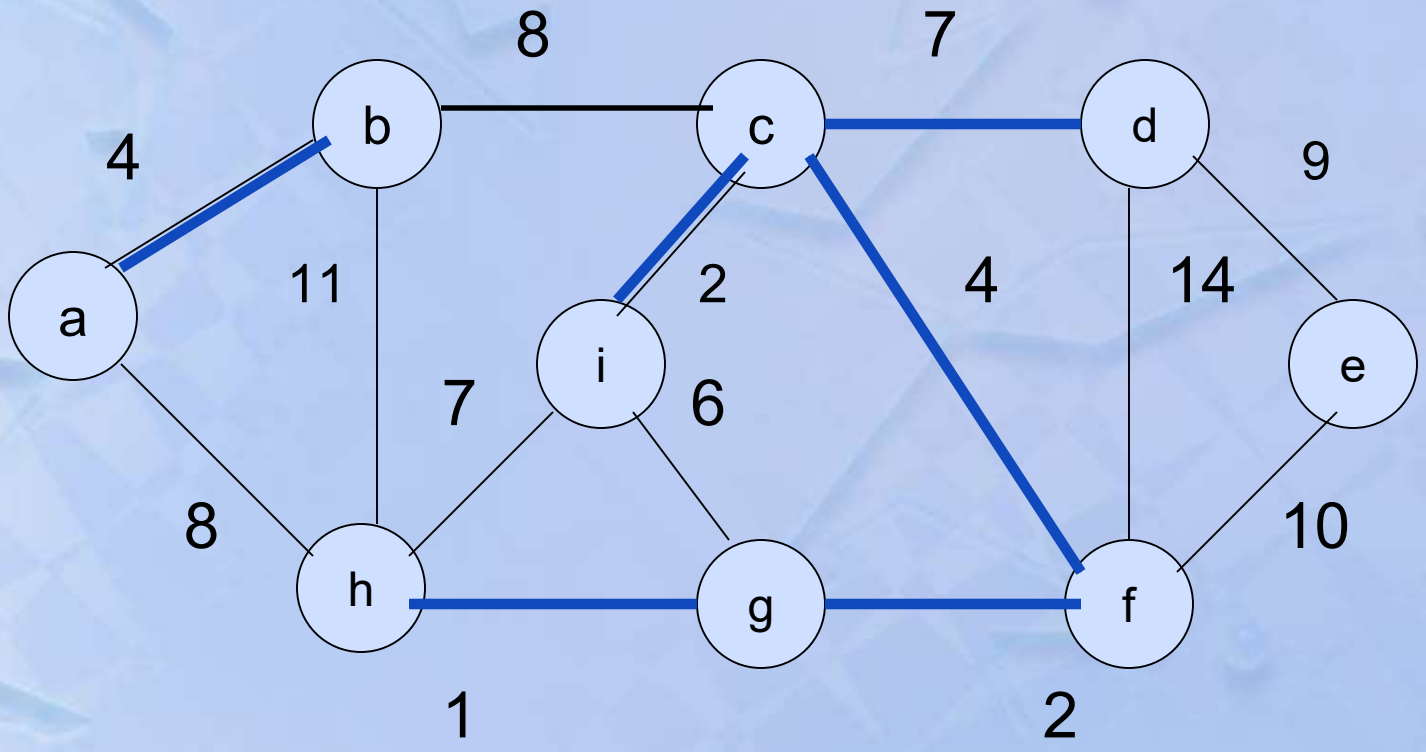
Step - 2



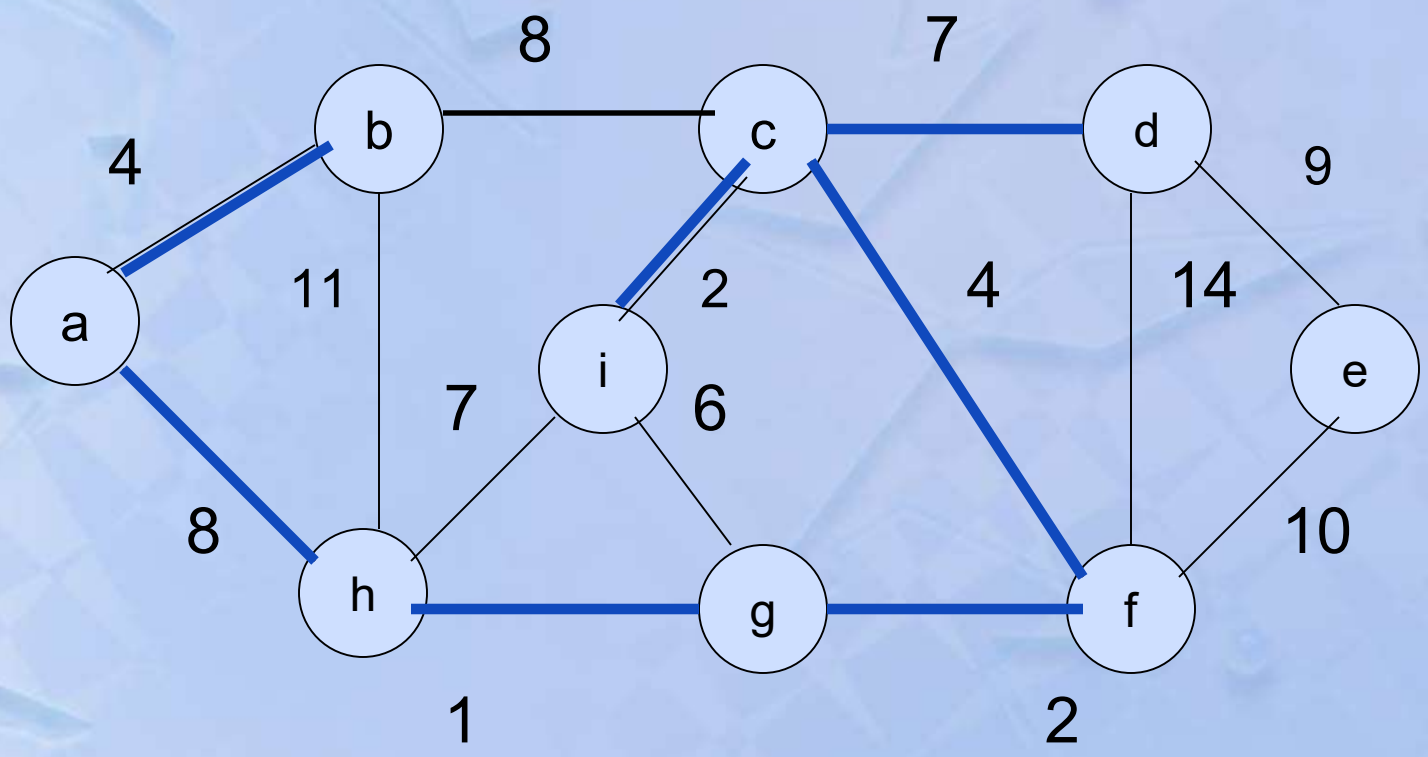
Step -3



Step -4



Step -5



Step -6