Design and Analysis of Algorithm

Greedy Methods
(Minimum Spanning Tree)

Lecture – 65-66

Overview

- A greedy algorithm always makes the choice that looks best at the moment. (i.e. it makes a locally optimal choice in the hope that this choice will lead to a globally optimal solution).
- The objective of this section is to explores optimization problems that are solvable by greedy algorithms.

Greedy Algorithm

- In mathematics, computer science and economics, an optimization problem is the problem of finding the best solution from all feasible solutions.
- Algorithms for optimization problems typically go through a sequence of steps, with a set of choices at each step.
- Many optimization problems can be solved using a greedy approach.
- Greedy algorithms are simple and straightforward.

Greedy Algorithm

- A greedy algorithm always makes the choice that looks best at the moment.
- That is, it makes a locally optimal choice in the hope that this choice will lead to a globally optimal solution.
- Greedy algorithms do not always yield optimal solutions, but for many problems they do.
- This algorithms are easy to invent, easy to implement and most of the time provides best and optimized solution.

Greedy Algorithm

- Application of Greedy Algorithm:
  - A simple but nontrivial problem, the activity-selection problem, for which a greedy algorithm efficiently computes a solution.
  - In combinatorics,(a branch of mathematics), a 'matroid' is a structure that abstracts and generalizes the notion of linear independence in vector spaces. Greedy algorithm always produces an optimal solution for such problems. Scheduling unit-time tasks with deadlines and penalties is an example of such problem.

Greedy Algorithm

- Application of Greedy Algorithm:
  - An important application of greedy techniques is the design of data-compression codes (i.e. Huffman code) .
  - The greedy method is quite powerful and works well for a wide range of problems. They are:
    - Minimum-spanning-tree algorithms

      (Example: Prims and Kruskal algorithm)
    - Single Source Shortest Path.

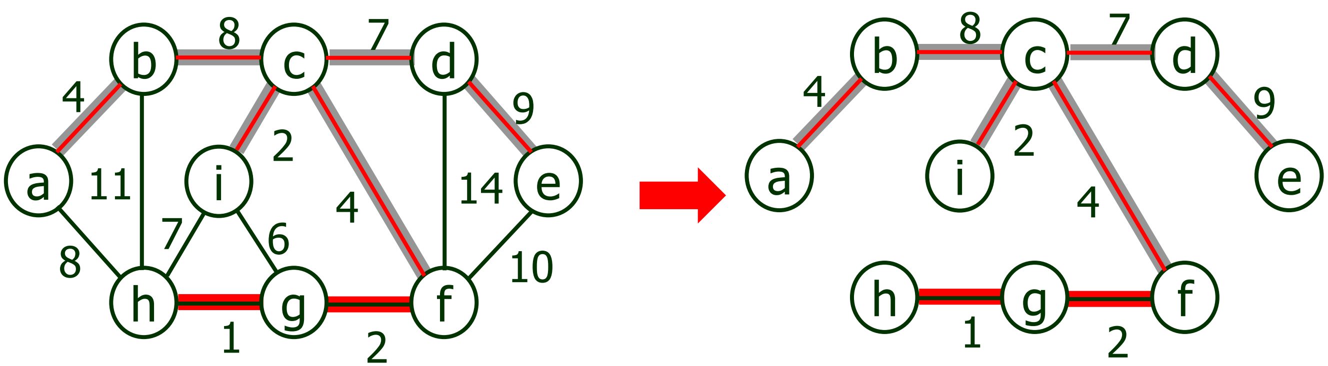
      (Example: Dijkstra's and Bellman ford algorithm)

Greedy Algorithm

- Application of Greedy Algorithm:
  - A problem exhibits optimal substructure if an optimal solution to the problem contains within it optimal solutions to subproblems.
  - This property is a key ingredient of assessing the applicability of dynamic programming as well as greedy algorithms.
  - The subtleties between the above two techniques are illustrated with the help of two variants of a classical optimization problem known as knapsack problem. These variants are:
    - 0-1 knapsack problem (Dynamic Programming)
    - Fractional knapsack problem (Greedy Algorithm)

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
  - Spanning Tree
    - A tree (i.e., connected, acyclic graph) which contains all the vertices of the graph
  - Minimum Spanning Tree
    - Spanning tree with the minimum sum of weights



Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
  - **Problem Definition**
    - A town has a set of houses and a set of roads.
    - A road connects 2 and only 2 houses.
    - A road connecting houses u and v has a repair cost $w(u, v)$
  - **Goal:**
    - Repair enough roads such that
      1. everyone stays connected: can reach every house from all other houses, and
      2. total repair cost is minimum.

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
  - **Graph model for MST**
    - Undirected graph $G = \langle V, E \rangle$
    - Weight $w(u, v)$ on each edge $(u, v) \in E$.
    - Find T ⊆ E such that
      1. T connects all vertices (T is a spanning tree), and
      2. $w(T) = \sum_{(u,v)\in T} w(u, v)$ is minimized.
  - **Properties of an MST**:
    - It has $|V| - 1$ edges.
    - It has no cycles.
    - It might not be unique

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
  - **Generic MST Algorithm**

GENERIC-MST (G, w)

$A = \emptyset$;

while A is not a spanning tree

find an edge (u, v) that is safe for A

$A = A \cup \{(u, v)\}$

return A

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
  - **Generic MST Algorithm**

GENERIC-MST (G, w)

$A = \emptyset$;

while A is not a spanning tree

find an edge (u, v) that is safe for A

$A = A \cup \{(u, v)\}$

return A

Use the loop invariant to show that this generic algorithm works.

- **Initialization:** The empty set trivially satisfies the loop invariant.
- **Maintenance:** Since we add only safe edges, A remains a subset of some MST.
- **Termination:** All edges added to A are in an MST, so when we stop, A is a spanning tree that is also an MST.

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
  - **Kruskal's Algorithm**
    - **Concept and Examples**
  - **Prim's Algorithm**
    - **Concept and Examples**

Greedy Algorithm

- Problem 4: Minimum Spanning Tree problem
  - Kruskal’s Algorithm
    - Concept and Examples
  - Prim’s Algorithm
    - Concept and Examples

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Kruskal’s Algorithm**
  - Kruskal’s Algorithm is a famous greedy algorithm.
  - It is used for finding the Minimum Spanning Tree (MST) of a given graph.
  - To apply Kruskal’s algorithm, the given graph must be weighted, connected and undirected.

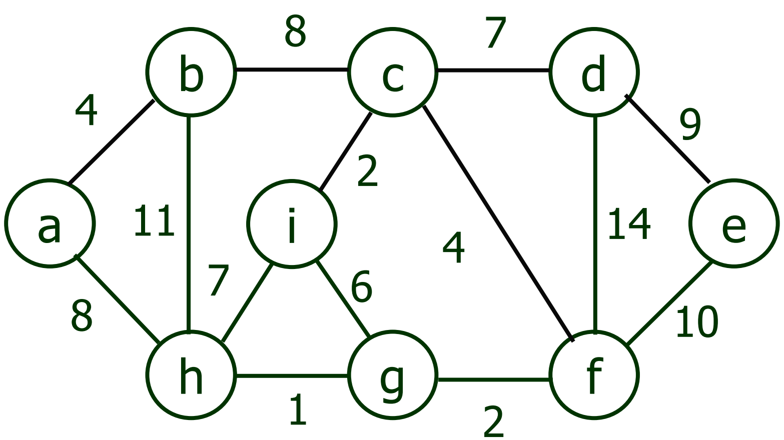
Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Kruskal's Algorithm**
  - Implementation of algorithm.
    - The implementation of Kruskal's Algorithm is explained in the following steps-
    - Step-01:
      - Sort all the edges from low weight to high weight.
    - Step-02:
      - Take the edge with the lowest weight and use it to connect the vertices of graph.
      - If adding an edge creates a cycle, then reject that edge and go for the next least weight edge.
    - Step-03:
      - Keep adding edges until all the vertices are connected and a Minimum Spanning Tree (MST) is obtained.

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Kruskal's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Kruskal's Algorithm-

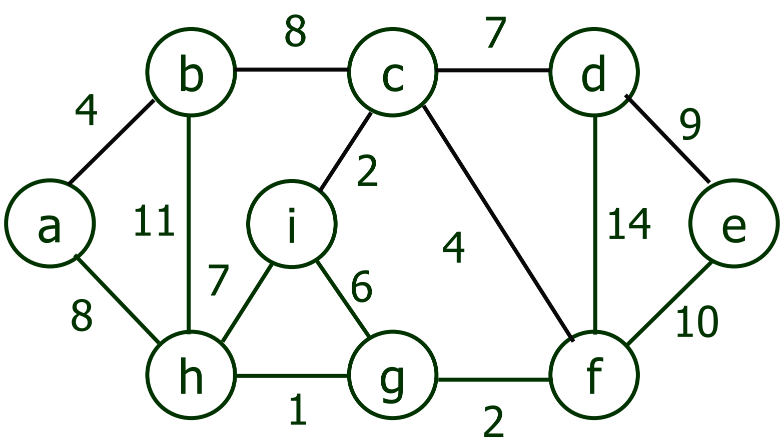


Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Kruskal's Algorithm:**

Example 1:

Solution: Read the edges

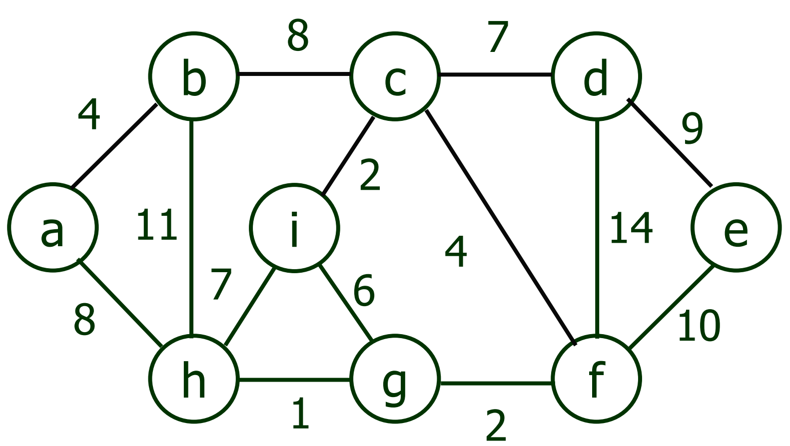


Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Kruskal's Algorithm:**

Example 1:

Solution: Read the edges



Edge	Weight
ab	4
bc	8
cd	7
de	9
ef	10
df	14
cf	4
ci	2
ih	7
ig	6
gf	2
gh	1
ah	8
bh	11

Greedy Algorithm

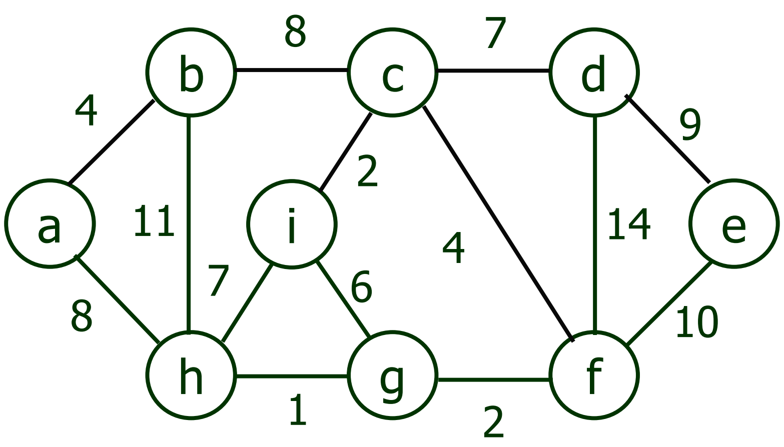
- **Problem 4: Minimum Spanning Tree problem**
- **Kruskal's Algorithm**
  - Implementation of algorithm.
    - The implementation of Kruskal's Algorithm is explained in the following steps-
    - Step-01:
      - Sort all the edges from low weight to high weight.
    - Step-02:
      - Take the edge with the lowest weight and use it to connect the vertices of graph.
      - If adding an edge creates a cycle, then reject that edge and go for the next least weight edge.
    - Step-03:
      - Keep adding edges until all the vertices are connected and a Minimum Spanning Tree (MST) is obtained.

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Kruskal's Algorithm:**

Example 1:

Solution: Apply Step 1



Edge	Weight
gh	1
ci	2
gf	2
ab	4
cf	4
ig	6
cd	7
ih	7
bc	8
ah	8
de	9
ef	10
bh	11
df	14

Greedy Algorithm

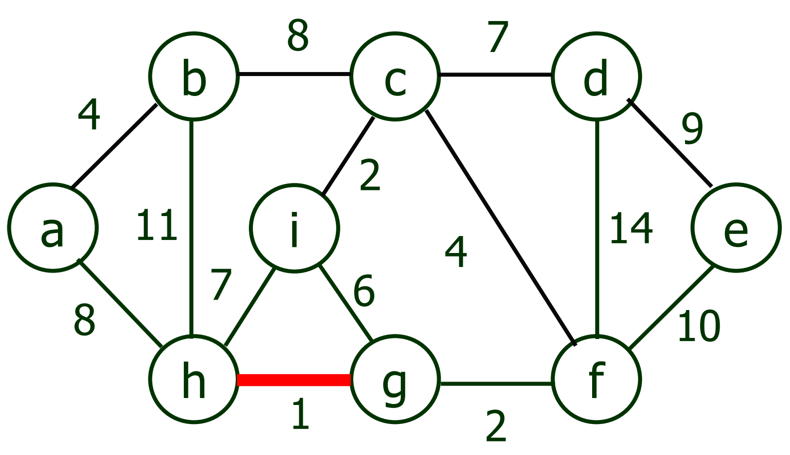
- **Problem 4: Minimum Spanning Tree problem**
- **Kruskal's Algorithm**
  - Implementation of algorithm.
    - The implementation of Kruskal's Algorithm is explained in the following steps-
    - Step-01:
      - Sort all the edges from low weight to high weight.
    - Step-02:
      - Take the edge with the lowest weight and use it to connect the vertices of graph.
      - If adding an edge creates a cycle, then reject that edge and go for the next least weight edge.
    - Step-03:
      - Keep adding edges until all the vertices are connected and a Minimum Spanning Tree (MST) is obtained.

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Kruskal's Algorithm:**

Example 1:

Solution: Apply Step 2 and 3



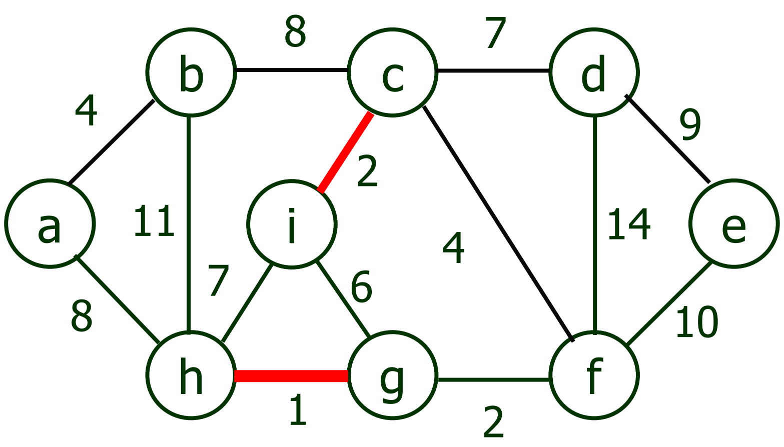
Edge	Weight
gh	1
ci	2
gf	2
ab	4
cf	4
ig	6
cd	7
ih	7
bc	8
ah	8
de	9
ef	10
bh	11
df	14

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Kruskal’s Algorithm:**

Example 1:

Solution: Apply Step 2 and 3



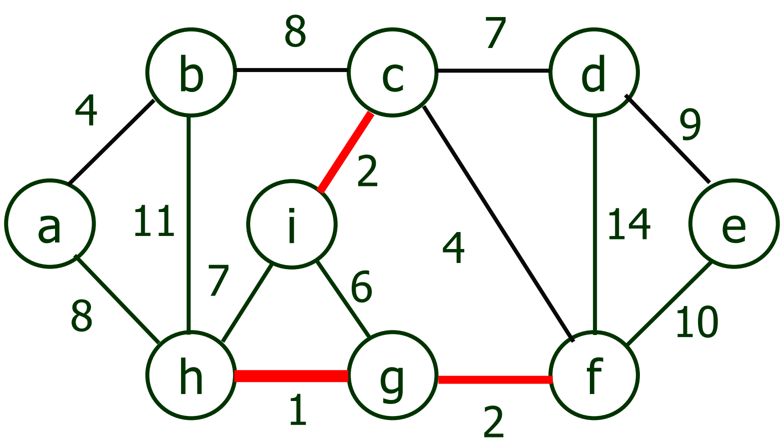
Edge	Weight
gh	1
ci	2
gf	2
ab	4
cf	4
ig	6
cd	7
ih	7
bc	8
ah	8
de	9
ef	10
bh	11
df	14

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Kruskal's Algorithm:**

Example 1:

Solution: Apply Step 2 and 3



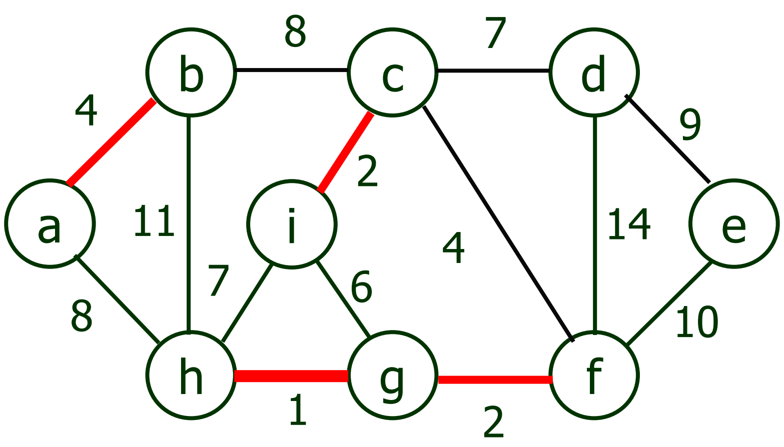
Edge	Weight
gh	1
ci	2
gf	2
ab	4
cf	4
ig	6
cd	7
ih	7
bc	8
ah	8
de	9
ef	10
bh	11
df	14

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Kruskal's Algorithm:**

Example 1:

Solution: Apply Step 2 and 3



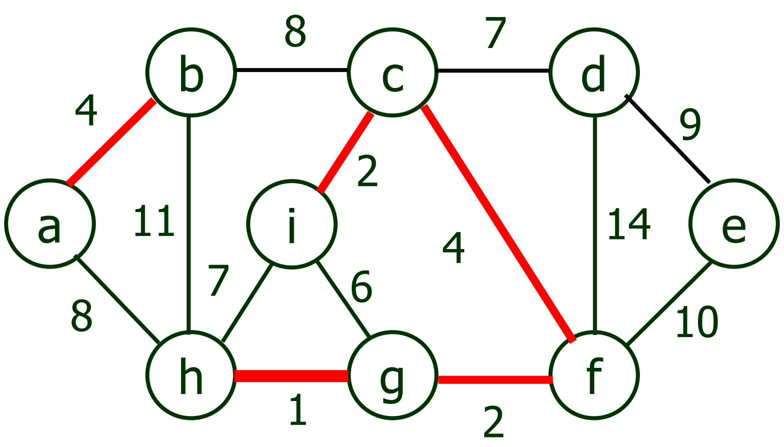
Edge	Weight
gh	1
ci	2
gf	2
ab	4
cf	4
ig	6
cd	7
ih	7
bc	8
ah	8
de	9
ef	10
bh	11
df	14

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Kruskal's Algorithm:**

Example 1:

Solution: Apply Step 2 and 3



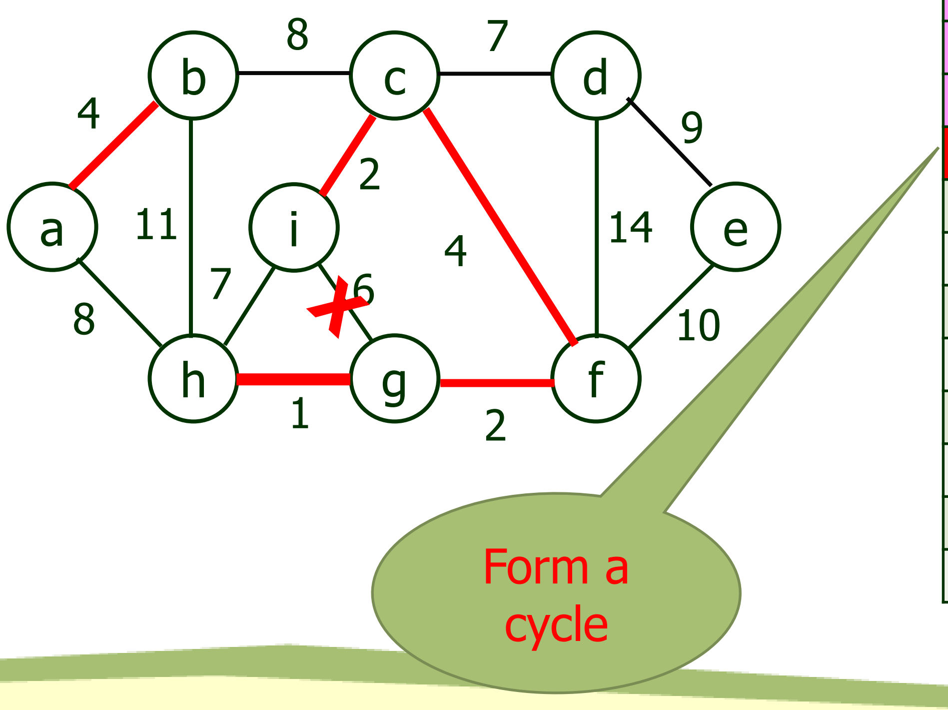
Edge	Weight
gh	1
ci	2
gf	2
ab	4
cf	4
ig	6
cd	7
ih	7
bc	8
ah	8
de	9
ef	10
bh	11
df	14

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Kruskal's Algorithm:**

Example 1:

Solution: Apply Step 2 and 3



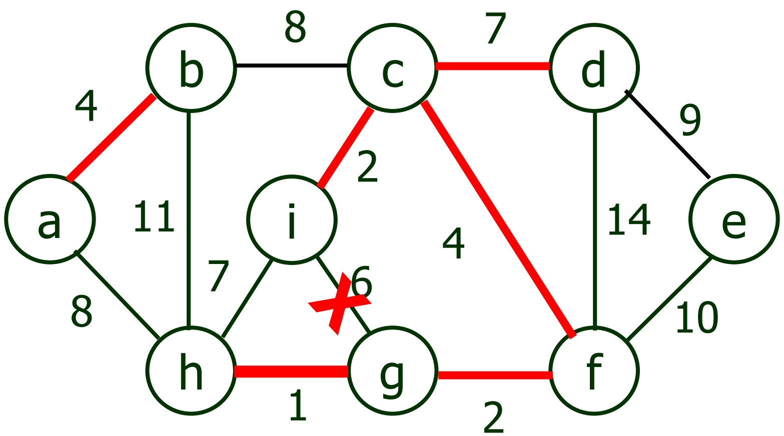
Edge	Weight
gh	1
ci	2
gf	2
ab	4
cf	4
ig	6
cd	7
ih	7
bc	8
ah	8
de	9
ef	10
bh	11
df	14

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Kruskal's Algorithm:**

Example 1:

Solution: Apply Step 2 and 3



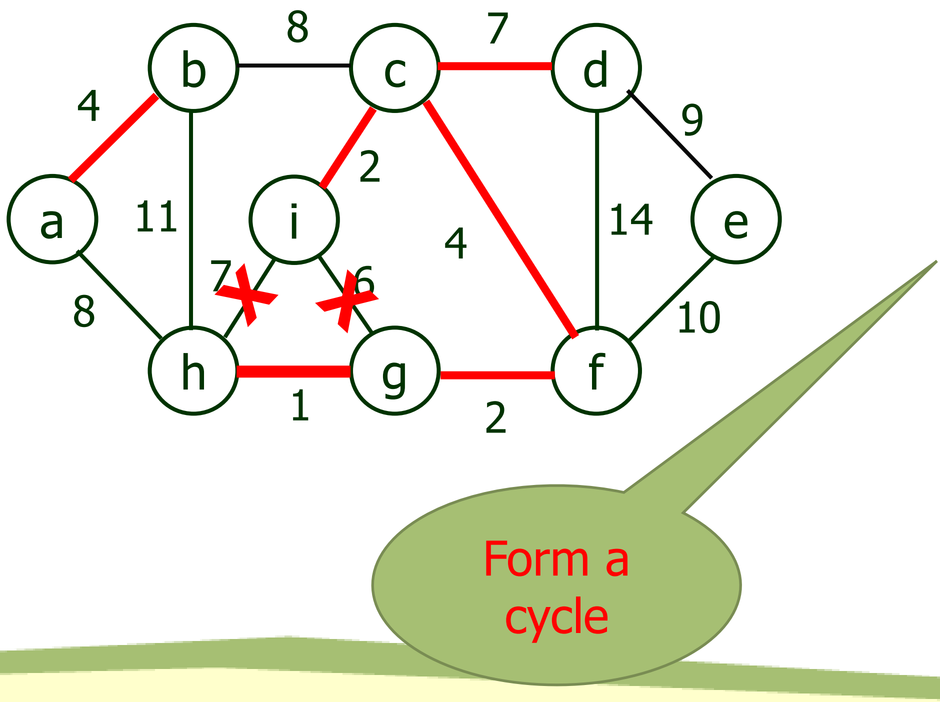
Edge	Weight
gh	1
ci	2
gf	2
ab	4
cf	4
ig	6
cd	7
ih	7
bc	8
ah	8
de	9
ef	10
bh	11
df	14

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Kruskal's Algorithm:**

Example 1:

Solution: Apply Step 2 and 3



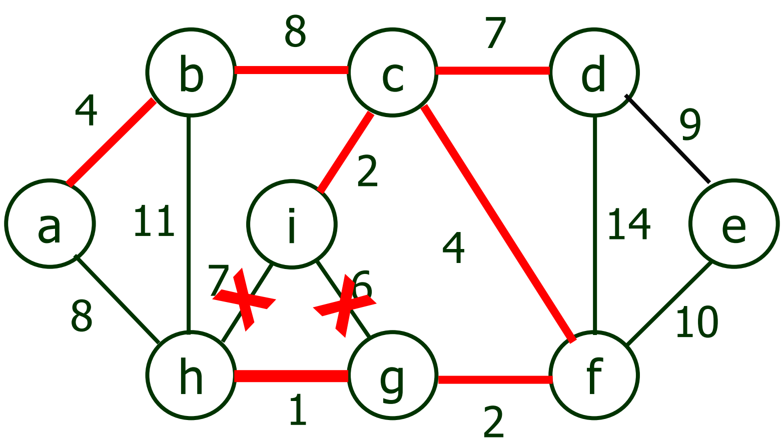
Edge	Weight
gh	1
ci	2
gf	2
ab	4
cf	4
ig	6
cd	7
ih	7
bc	8
ah	8
de	9
ef	10
bh	11
df	14

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Kruskal's Algorithm:**

Example 1:

Solution: Apply Step 2 and 3



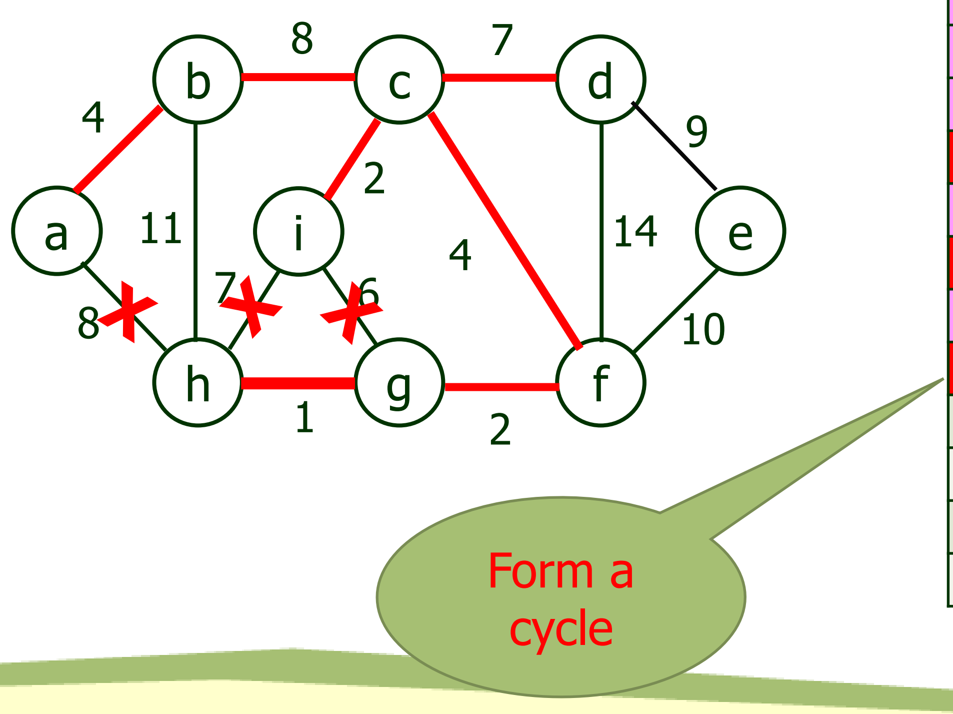
Edge	Weight
gh	1
ci	2
gf	2
ab	4
cf	4
ig	6
cd	7
ih	7
bc	8
ah	8
de	9
ef	10
bh	11
df	14

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Kruskal's Algorithm:**

Example 1:

Solution: Apply Step 2 and 3



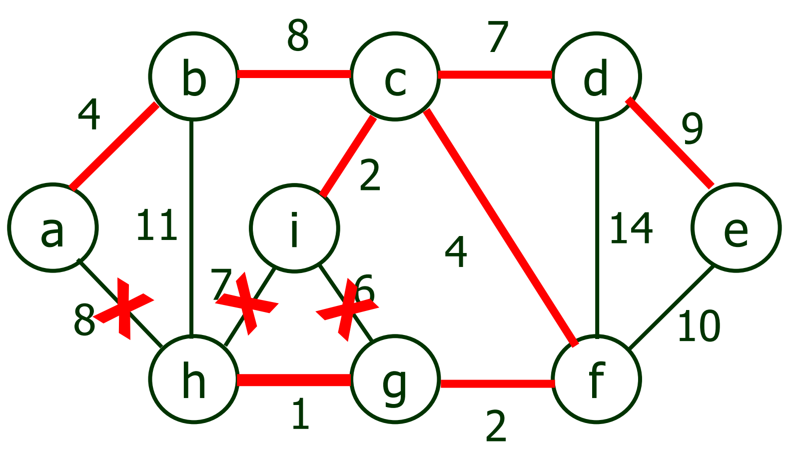
Edge	Weight
gh	1
ci	2
gf	2
ab	4
cf	4
ig	6
cd	7
ih	7
bc	8
ah	8
de	9
ef	10
bh	11
df	14

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Kruskal's Algorithm:**

Example 1:

Solution: Apply Step 2 and 3



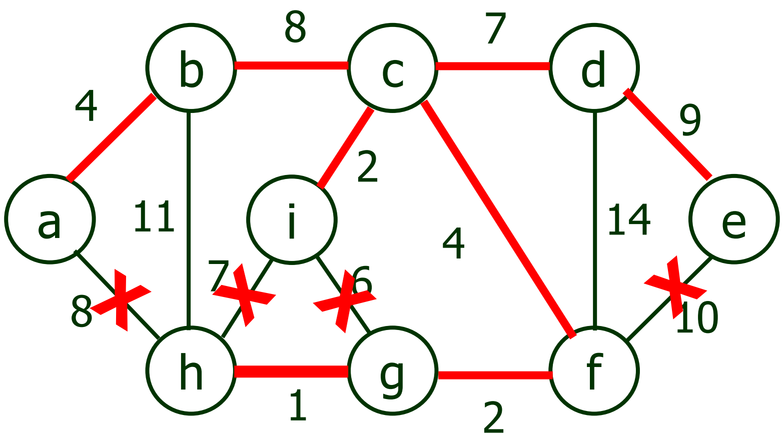
Edge	Weight
gh	1
ci	2
gf	2
ab	4
cf	4
ig	6
cd	7
ih	7
bc	8
ah	8
de	9
ef	10
bh	11
df	14

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
  - **Kruskal's Algorithm:**

Example 1:

Solution: Apply Step 2 and 3



Edge	Weight
gh	1
ci	2
gf	2
ab	4
cf	4
ig	6
cd	7
ih	7
bc	8
ah	8
de	9
ef	10
bh	11
df	14

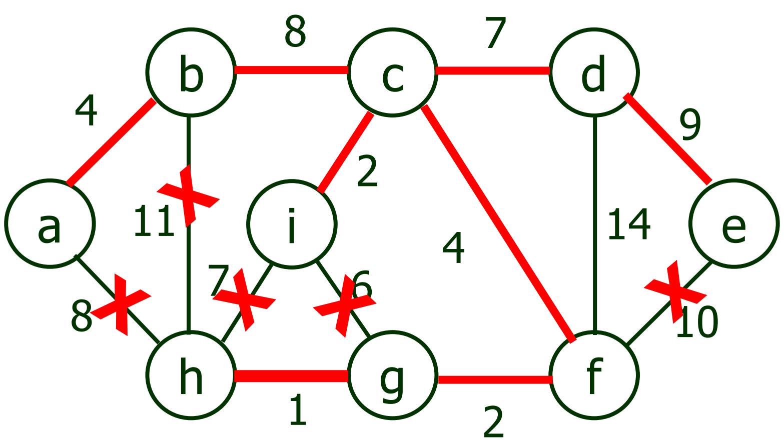
Form a cycle

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Kruskal's Algorithm:**

Example 1:

Solution: Apply Step 2 and 3



Edge	Weight
gh	1
ci	2
gf	2
ab	4
cf	4
ig	6
cd	7
ih	7
bc	8
ah	8
de	9
ef	10
bh	11
df	14

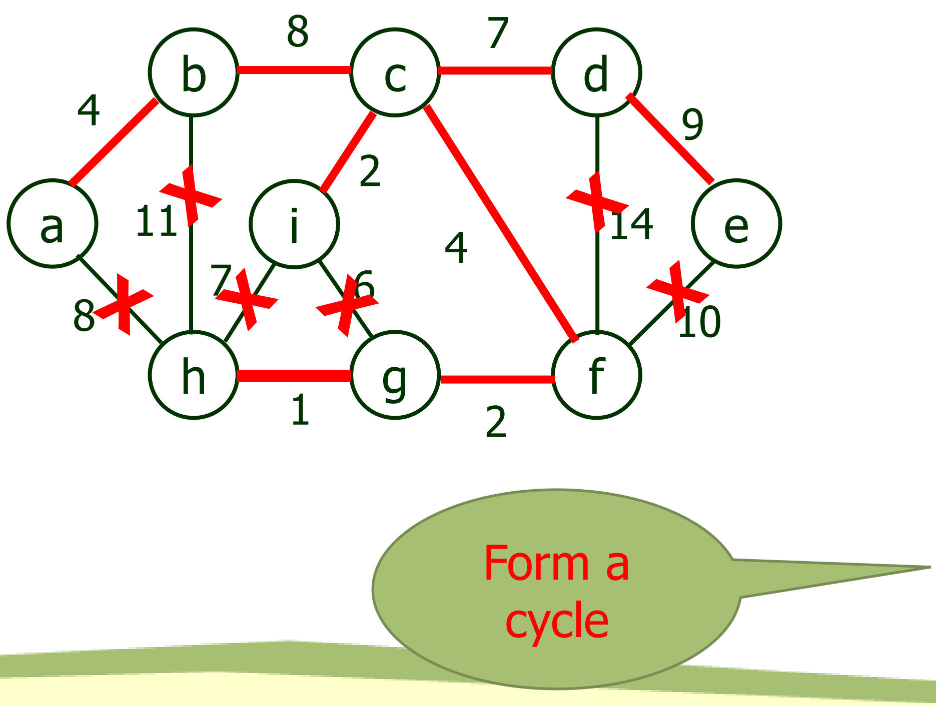
Form a cycle

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Kruskal's Algorithm:**

Example 1:

Solution: Apply Step 2 and 3



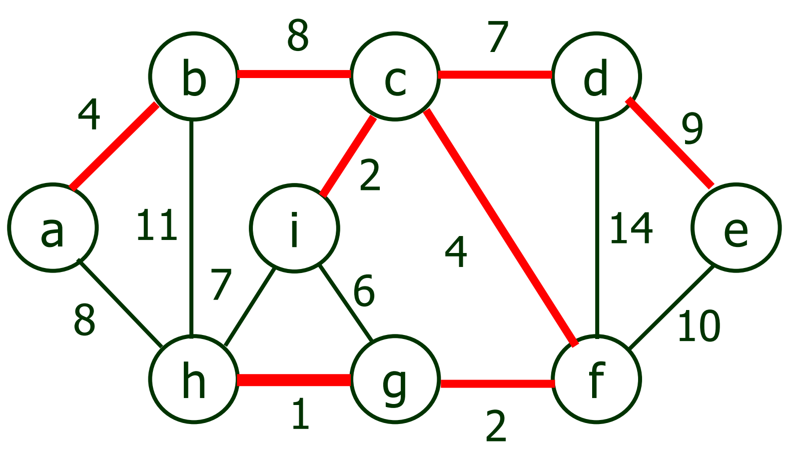
Edge	Weight
gh	1
ci	2
gf	2
ab	4
cf	4
ig	6
cd	7
ih	7
bc	8
ah	8
de	9
ef	10
bh	11
df	14

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Kruskal's Algorithm:**

Example 1:

Solution: Apply Step 2 and 3



Edge	Weight
gh	1
ci	2
gf	2
ab	4
cf	4
ig	6
cd	7
ih	7
bc	8
ah	8
de	9
ef	10
bh	11
df	14

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Kruskal's Algorithm**

MST-KRUSKAL(G, w)

```
A ← Ø
for each vertex v  V[G]
    do MAKE-SET(v)
sort the edges of E into nondecreasing order by weight w
for each edge (u, v)  E, taken in nondecreasing order by weight
    do if FIND-SET(u) ≠ FIND-SET(v)
        then A ← A  {(u, v)}
            UNION(u, v)
return A
```

$O(V)$

$O(E \lg E)$

$O(E)$

Hence the total Time Complexity is $O(V) + O(E \lg E) + O(E)$

$O(V + E + E \lg E) = O(E \lg E)$

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Kruskal's Algorithm(Analysis)**
  - The running time of Kruskal's algorithm for a graph $G = (V, E)$ depends on the implementation of the disjoint-set data structure.
  - Initializing the set A in line 1 takes $O(1)$ time, and the time to sort the edges in line 4 is O(E lg E).
  - The for loop of lines 5-8 performs $O(E)$ FIND-SET and UNION operations on the disjoint-set forest. Along with the |V| MAKE-SET operations, these take a total of $O(V + E)$ time.

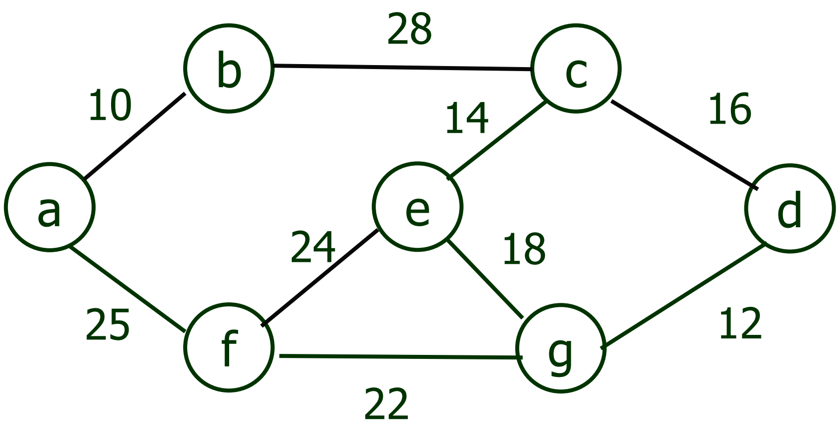
Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Kruskal's Algorithm (Analysis)**
  - the total running time of Kruskal's algorithm is $O(E \lg E)$.
  - Observing that $|E| < |V|^2$ (if the graph is a dense graph)
  - Apply log both side $\Longrightarrow \lg |E| = O(\lg V)$
  - Hence the running time of Kruskal's algorithm as $O(E \lg V)$.

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Kruskal's Algorithm:**

Example 2: Construct the minimum spanning tree (MST) for the given graph using Kruskal's Algorithm-



Self Practice

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
  - **Kruskal's Algorithm**
    - **Concept and Examples**
  - **Prim's Algorithm**
    - **Concept and Examples**

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm**
  - Prim's Algorithm is a famous greedy algorithm.
  - It is used for finding the Minimum Spanning Tree (MST) of a given graph.
  - To apply Prim's algorithm, the given graph must be weighted, connected and undirected.

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm (Implimentation)**
  - The implementation of Prim's Algorithm is explained in the following steps-
  - Step-1:
    - Randomly choose any vertex.
    - The vertex connecting to the edge having least weight is usually selected.

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm (Implementation)**
  - Step-2:
    - Find all the edges that connect the tree to new vertices.
    - Find the least weight edge among those edges and include it in the existing tree.
    - If including that edge creates a cycle, then reject that edge and look for the next least weight edge.
  - Step-03:
    - Keep repeating step-2 until all the vertices are included and Minimum Spanning Tree (MST) is obtained.

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm (Implementation)**
  - Step-2:
    - Find all the edges that connect the tree to new vertices.
    - Find the least weight edge among those edges and include it in the existing tree.
    - If including that edge creates a cycle, then reject that edge and look for the next least weight edge.
  - Step-03:
    - Keep repeating step-2 until all the vertices are included and Minimum Spanning Tree (MST) is obtained.

Greedy Algorithm

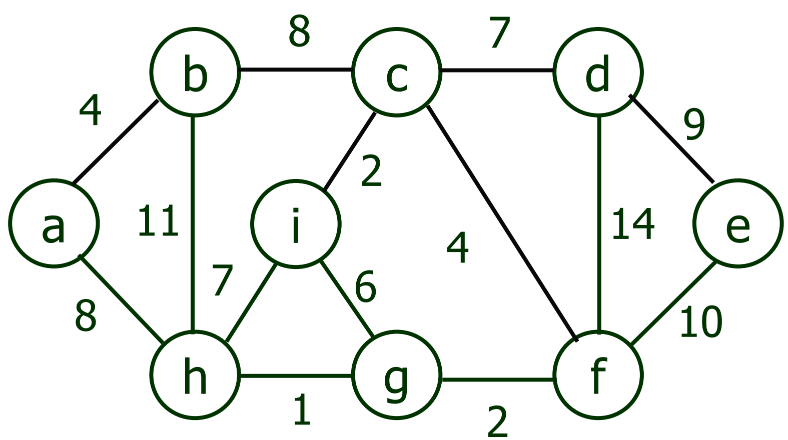
- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm: (Algorithm)**

```
MST-PRIM(G, w, r)
for each u ∈ V[G]
        do key[u] ← ∞
              π[u] ← NIL
key[r] ← 0
Q ← V [G]
while Q ≠ Ø
        do u ← EXTRACT-MIN(Q)
              for each v ∈ Adj[u]
                        do if v ∈ Q and w(u, v) < key[v]
                                        then п[v] ← u
                                                key[v] ← w(u, v)
```

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

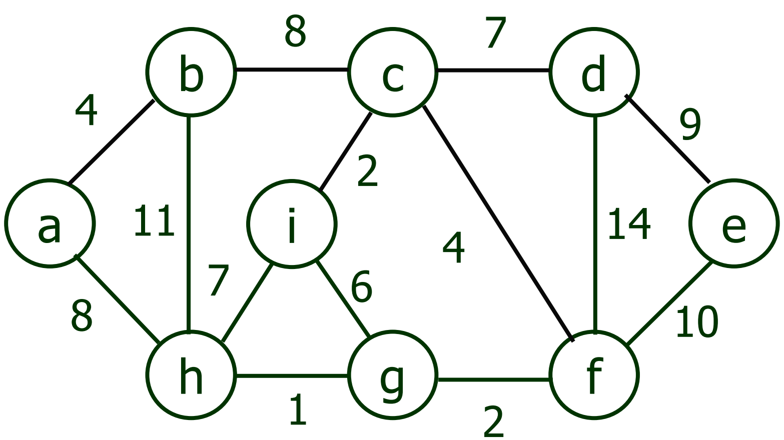
Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-

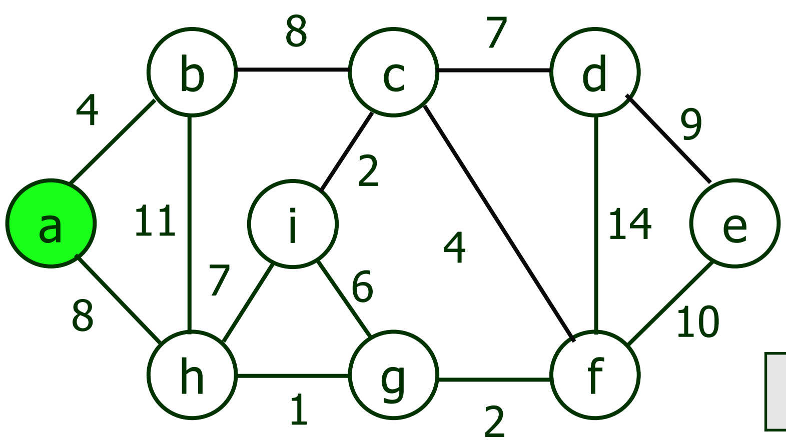


V	a	b	c	d	e	f	g	h	i
key	∞	∞	∞	∞	∞	∞	∞	∞	∞
Π	/	/	/	/	/	/	/	/	/

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 1
Assume that 'a' is the root.

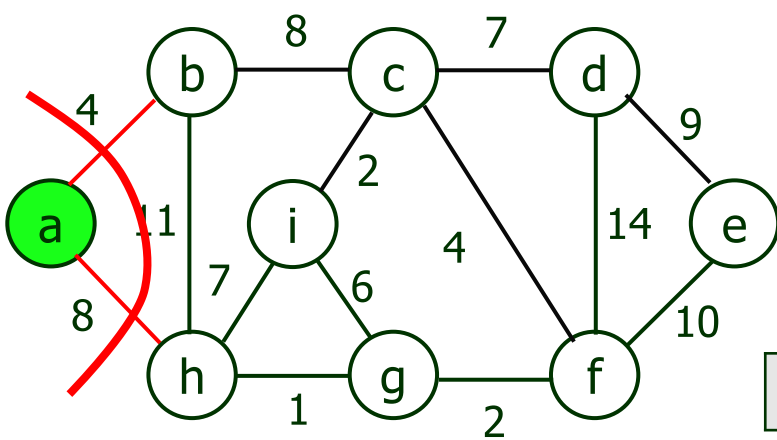
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	∞	∞	∞	∞	∞	∞	∞	∞
Π	/	/	/	/	/	/	/	/	/

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 1

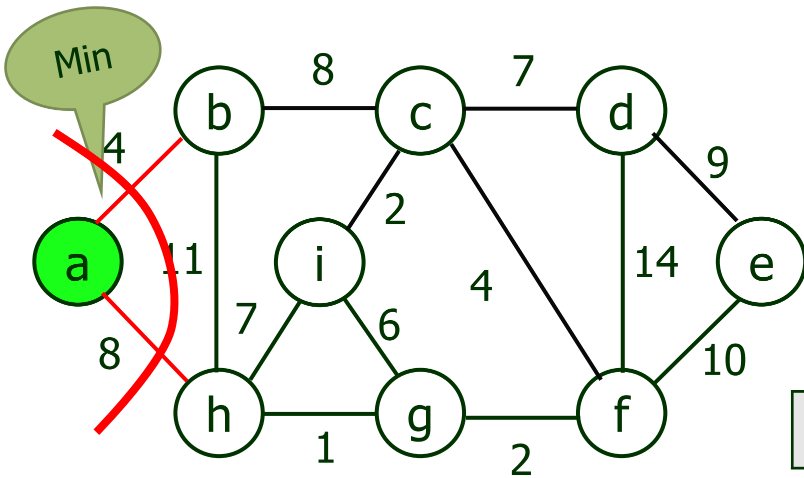
Q	a	b	c	d	e	f	g	h	i

V	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
key	0	4	∞	∞	∞	∞	∞	8	∞
Π	/	a	/	/	/	/	/	a	/

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim’s Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim’s Algorithm-



Use step 1

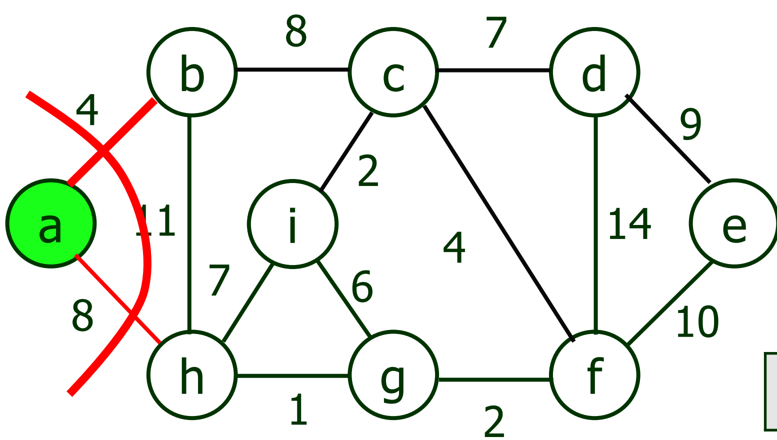
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	∞	∞	∞	∞	∞	8	∞
Π	/	a	/	/	/	/	/	a	/

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 1

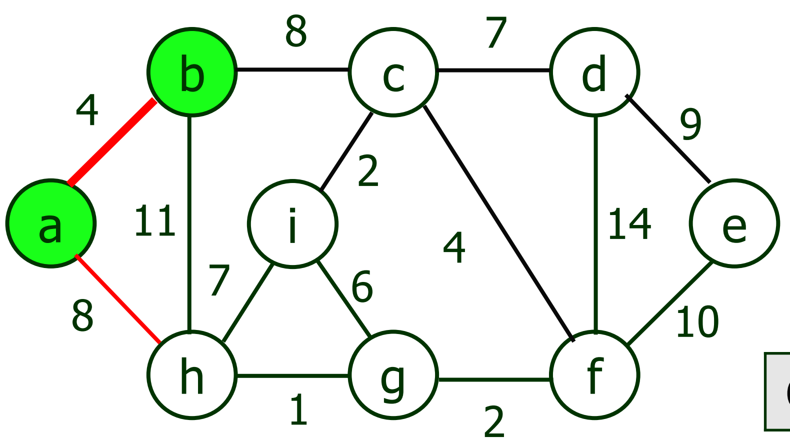
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	∞	∞	∞	∞	∞	8	∞
Π	/	a	/	/	/	/	/	a	/

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Kruskal's Algorithm-



Use step 2 and 3 until MST form

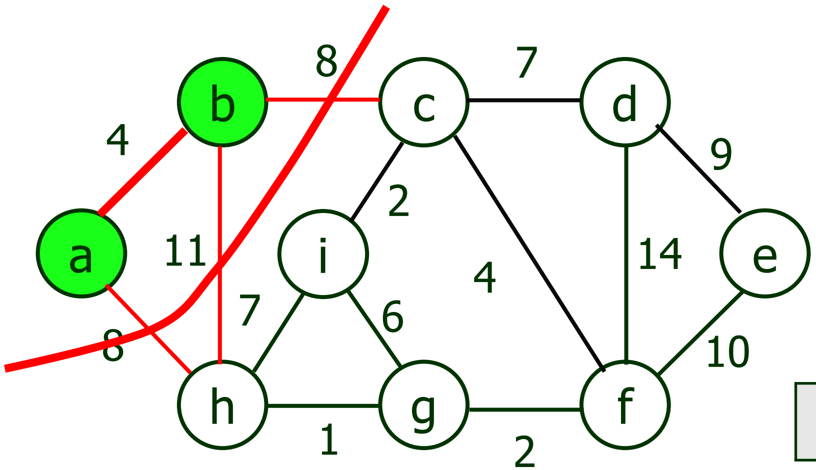
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	∞	∞	∞	∞	∞	8	∞
Π	/	a	/	/	/	/	/	a	/

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

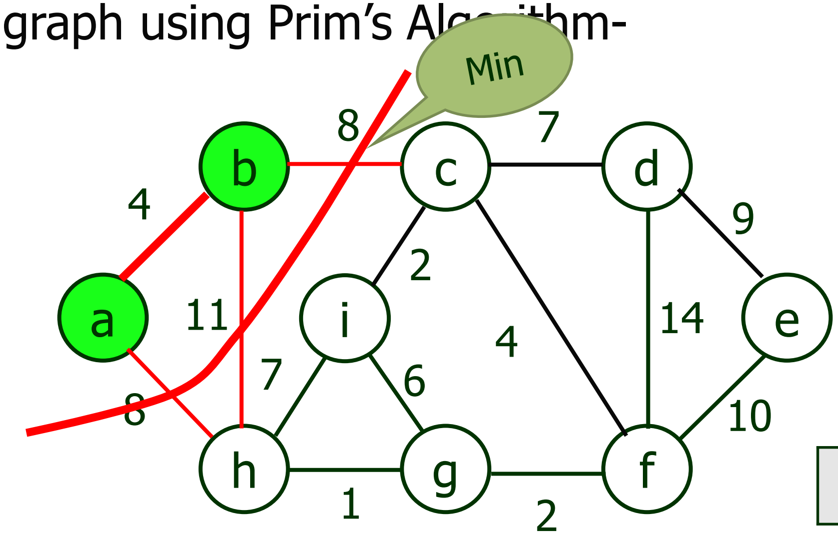
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	∞	∞	∞	∞	8	∞
Π	/	a	b	/	/	/	/	a	/

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

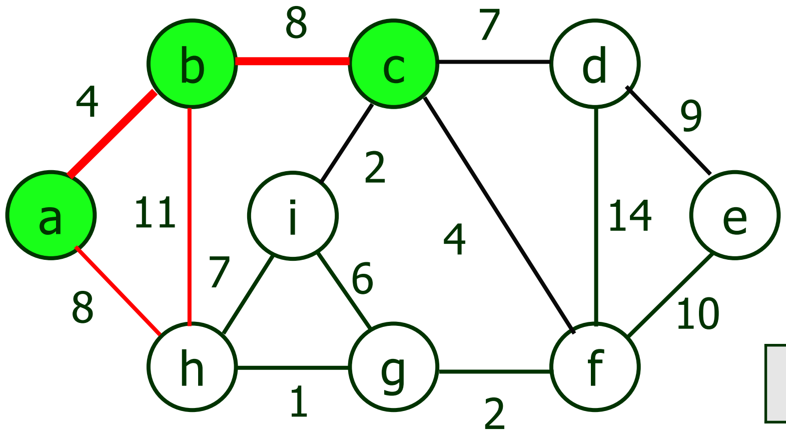
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	∞	∞	∞	∞	8	∞
Π	/	a	b	/	/	/	/	a	/

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

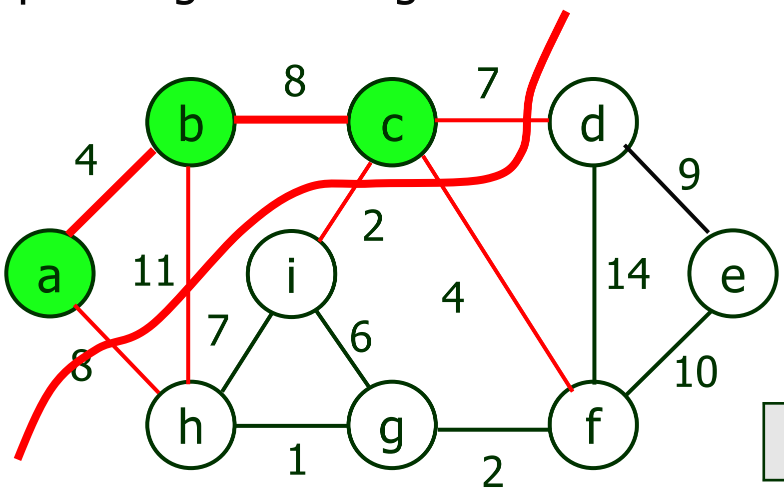
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	∞	∞	∞	∞	8	∞
Π	/	a	b	/	/	/	/	a	/

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

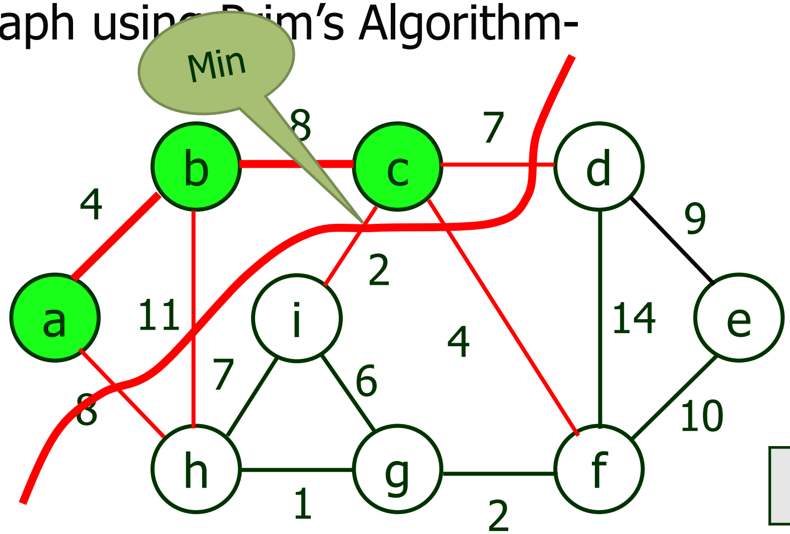
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	∞	4	∞	8	2
Π	/	a	b	c	/	c	/	a	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

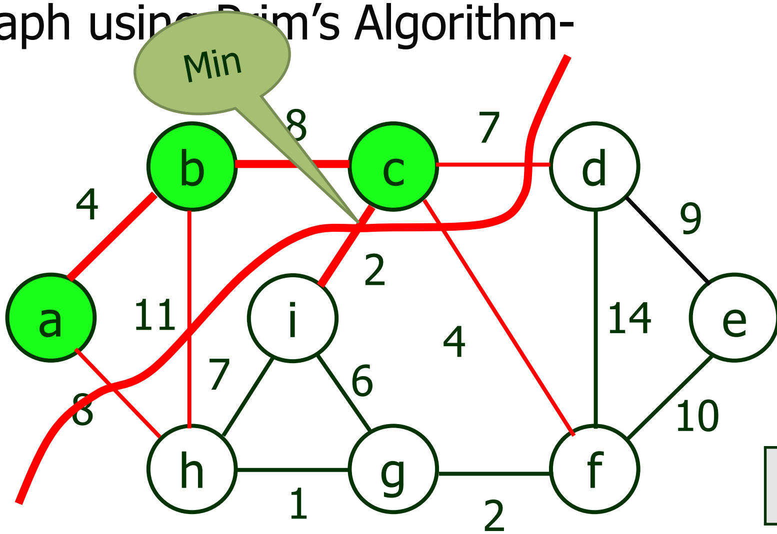
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	∞	4	∞	8	2
Π	/	a	b	c	/	c	/	a	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

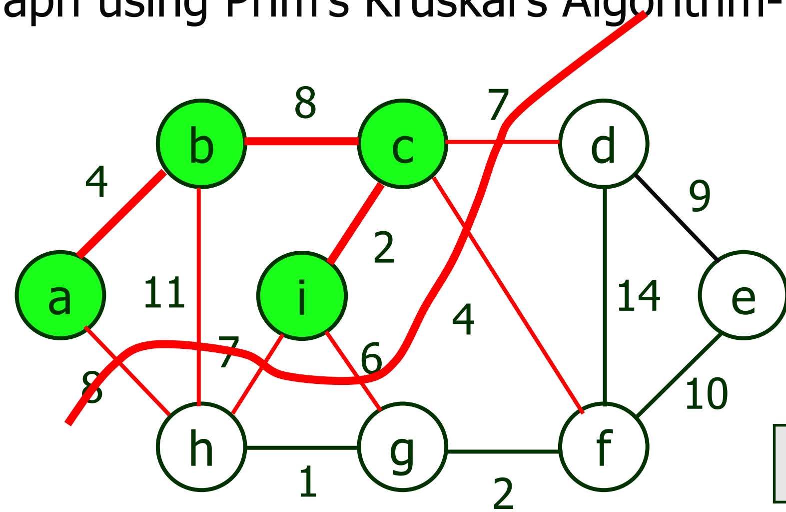
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	∞	4	∞	8	2
Π	/	a	b	c	/	c	/	a	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Kruskal's Algorithm-



Use step 2 and 3 until MST form

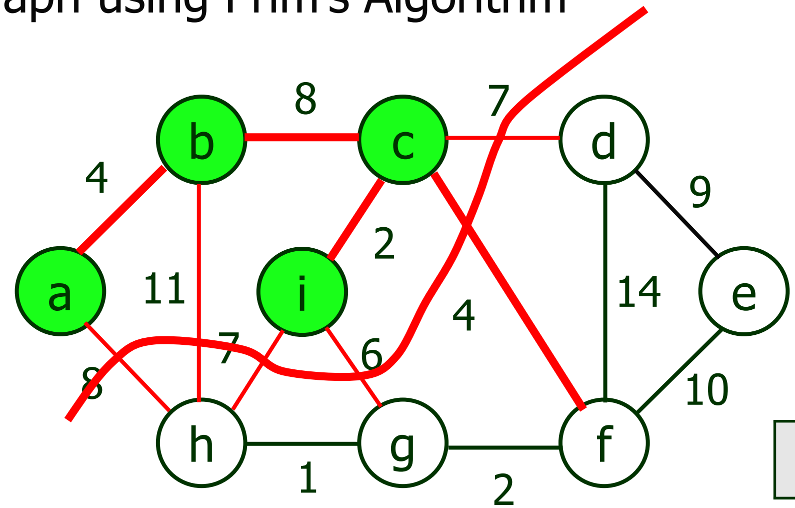
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	∞	4	6	7	2
Π	/	a	b	c	/	c	i	i	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

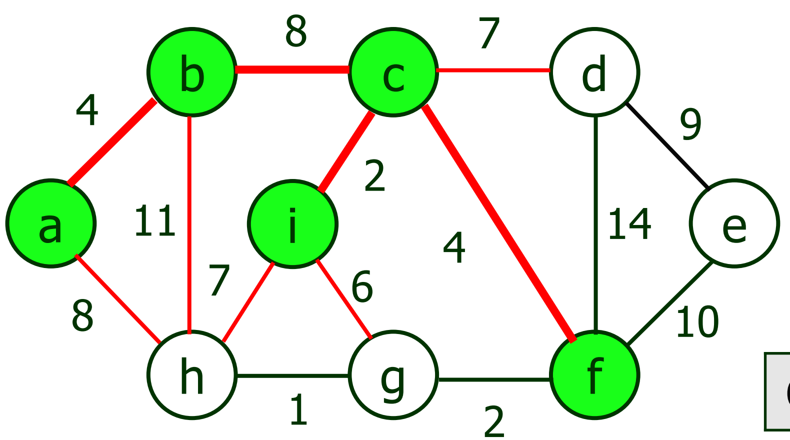
Q	a	b	c	d	e	f	g	h	i

V	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
key	0	4	8	7	∞	4	6	7	2
Π	/	a	b	c	/	c	i	i	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim’s Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim’s Algorithm-



Use step 2 and 3 until MST form

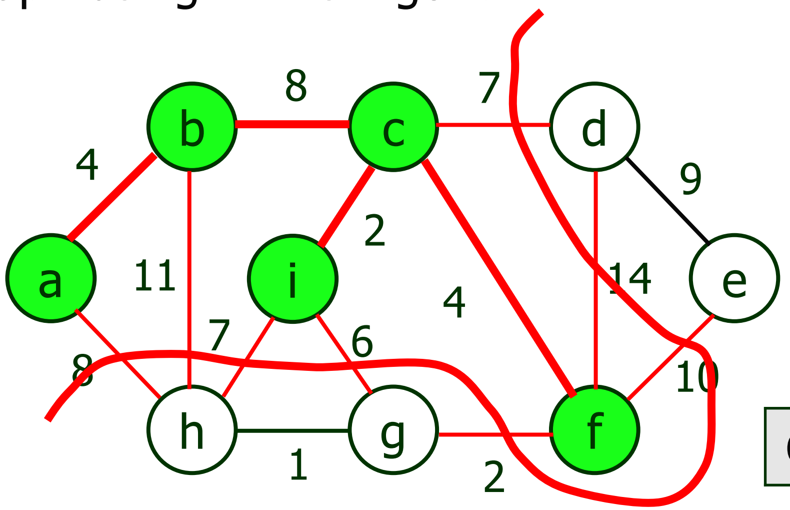
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	∞	4	6	7	2
Π	/	a	b	c	/	c	i	i	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

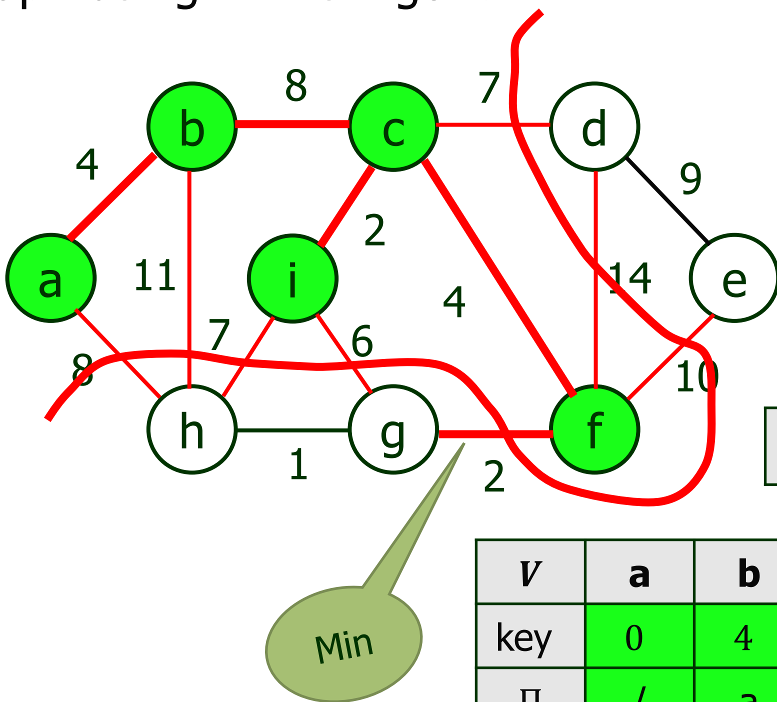
Q	a	b	c	d	e	f	g	h	i

V	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
key	0	4	8	7	10	4	2	7	2
Π	/	a	b	c	f	c	f	i	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim’s Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim’s Algorithm-



Use step 2 and 3 until MST form

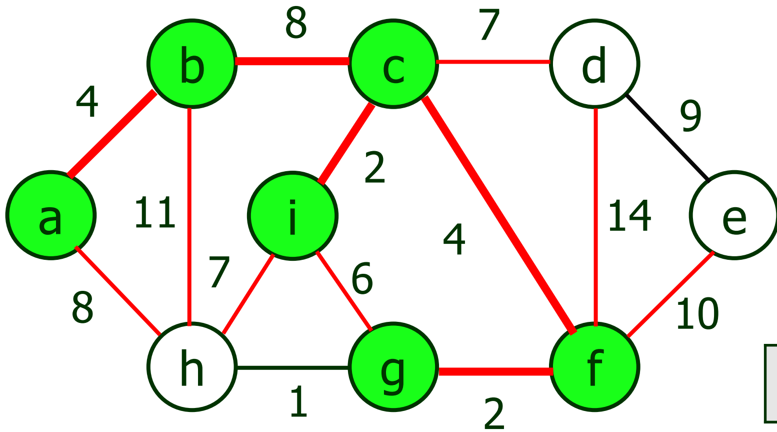
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	10	4	2	7	2
Π	/	a	b	c	f	c	f	i	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

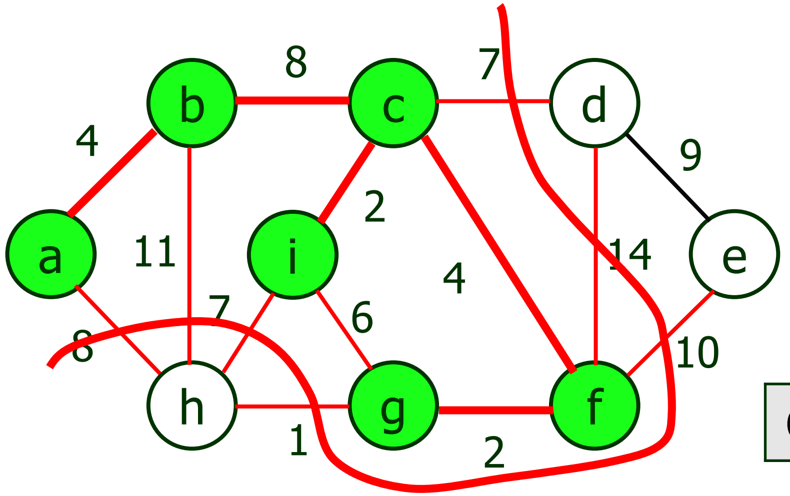
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	10	4	2	7	2
Π	/	a	b	c	f	c	f	i	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

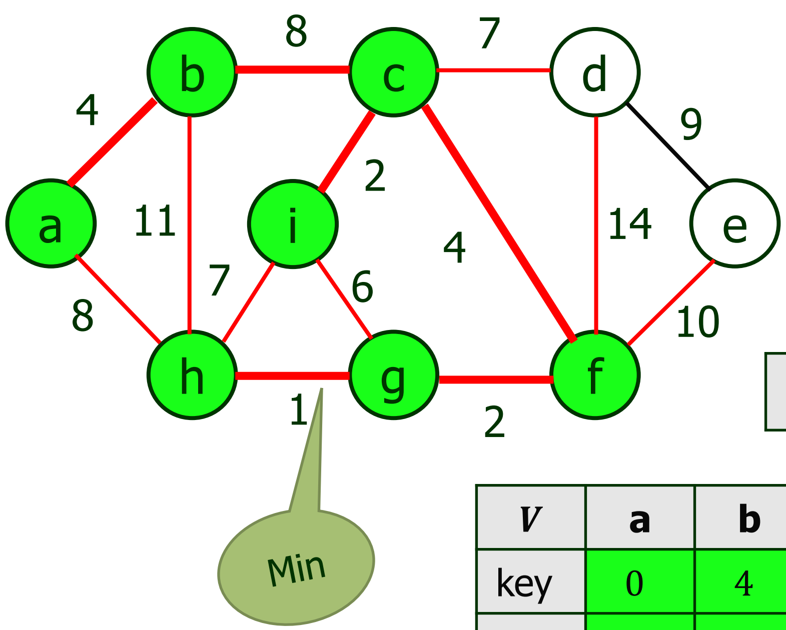
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	10	4	2	1	2
Π	/	a	b	c	f	c	f	g	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

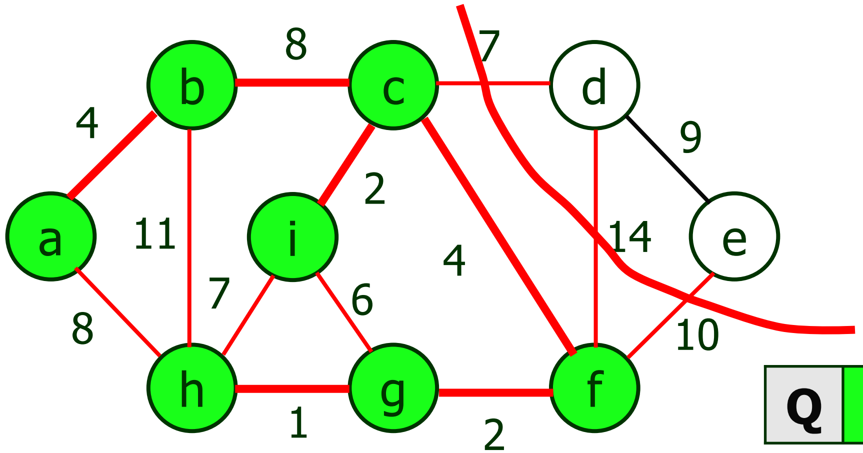
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	10	4	2	1	2
Π	/	a	b	c	f	c	f	h	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

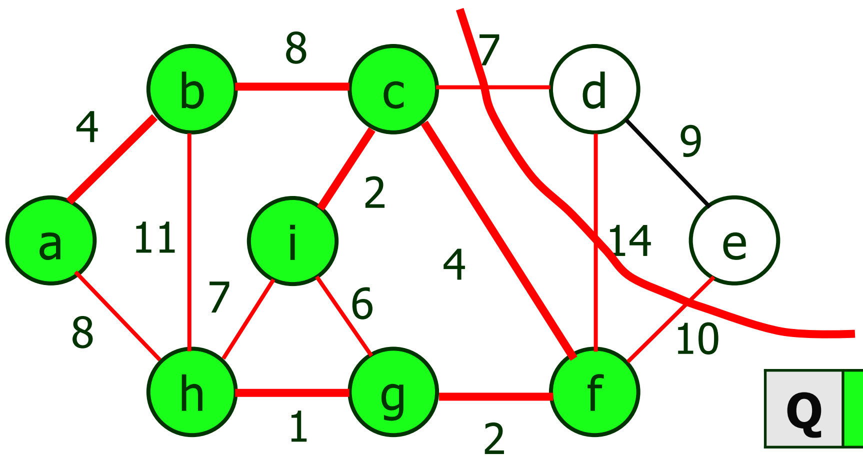
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	10	4	2	1	2
Π	/	a	b	c	f	c	f	h	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

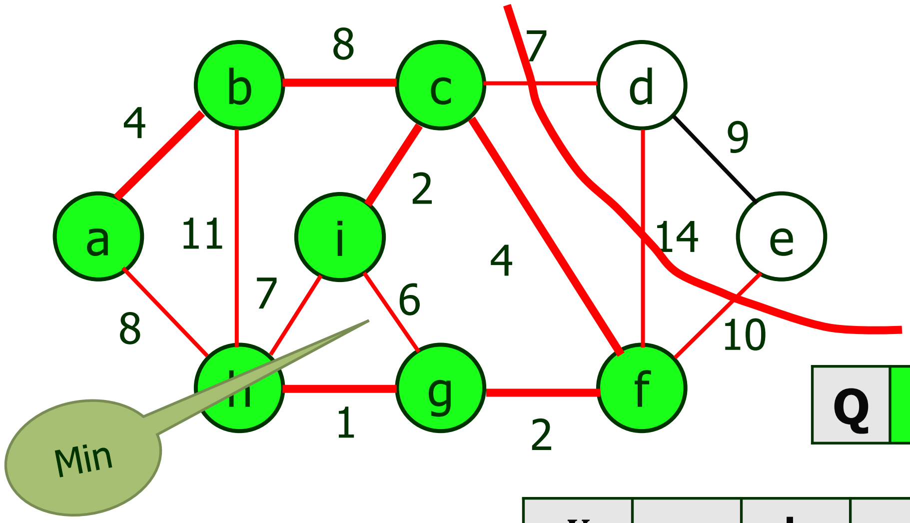
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	10	4	2	1	2
Π	/	a	b	c	f	c	f	h	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

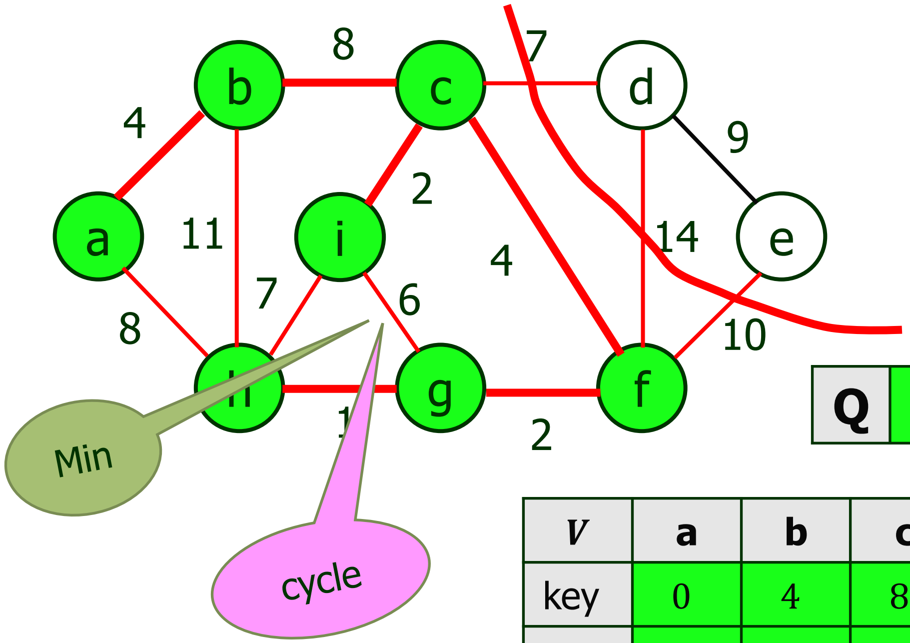
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	10	4	2	1	2
Π	/	a	b	c	f	c	f	h	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

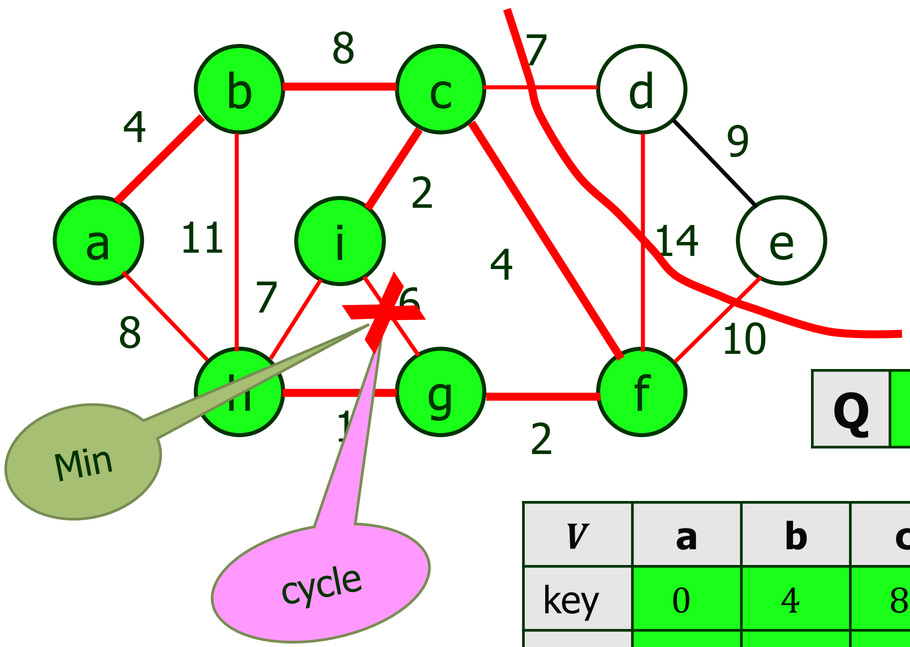
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	10	4	2	1	2
Π	/	a	b	c	f	c	f	h	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

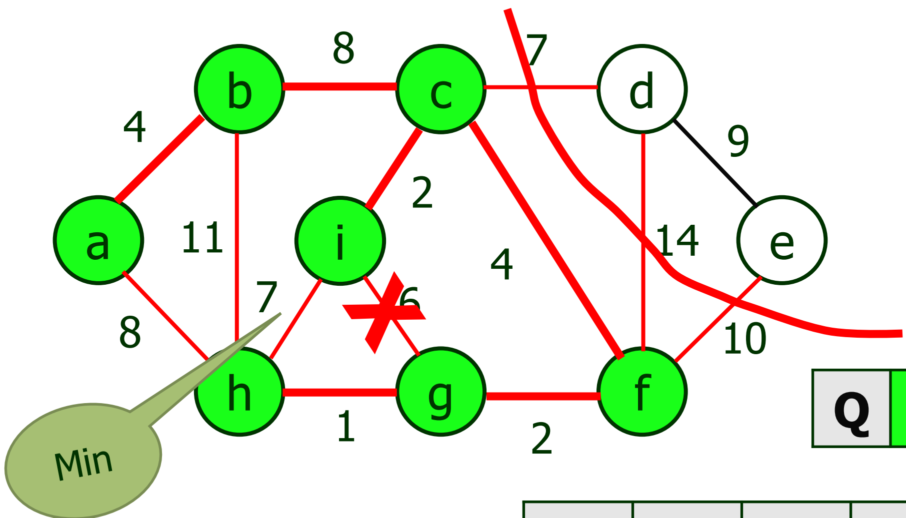
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	10	4	2	1	2
Π	/	a	b	c	f	c	f	h	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

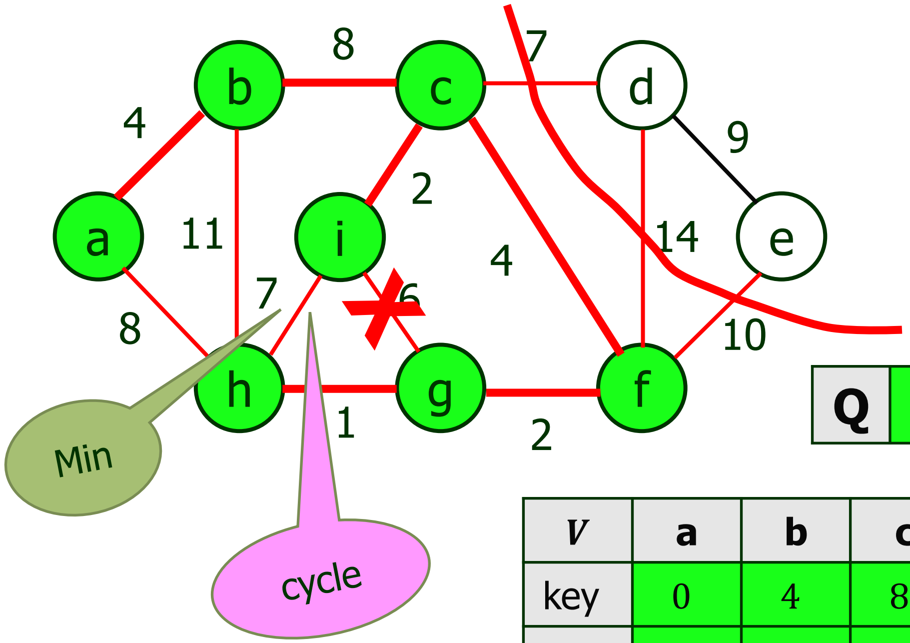
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	10	4	2	1	2
Π	/	a	b	c	f	c	f	h	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

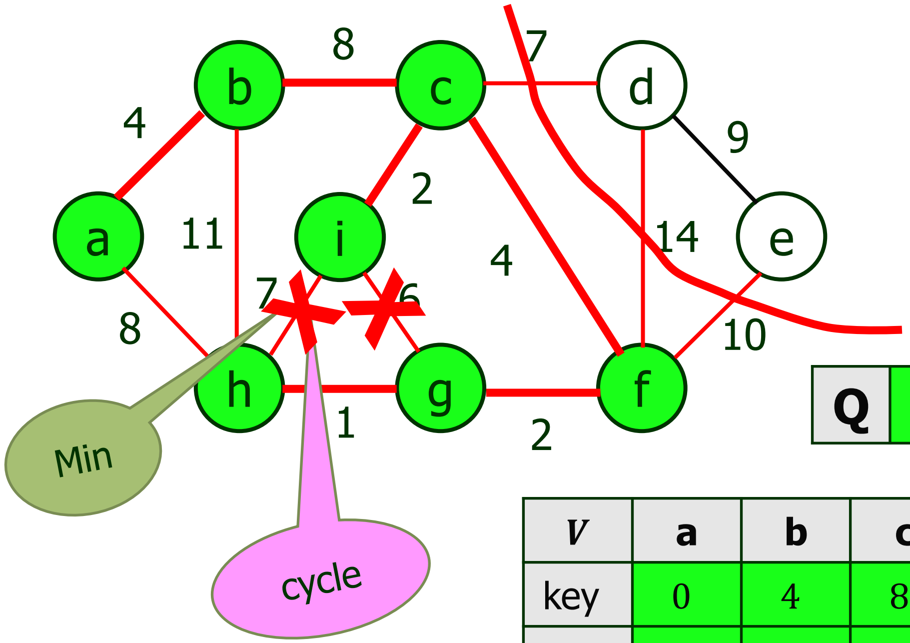
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	10	4	2	1	2
Π	/	a	b	c	f	c	f	h	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

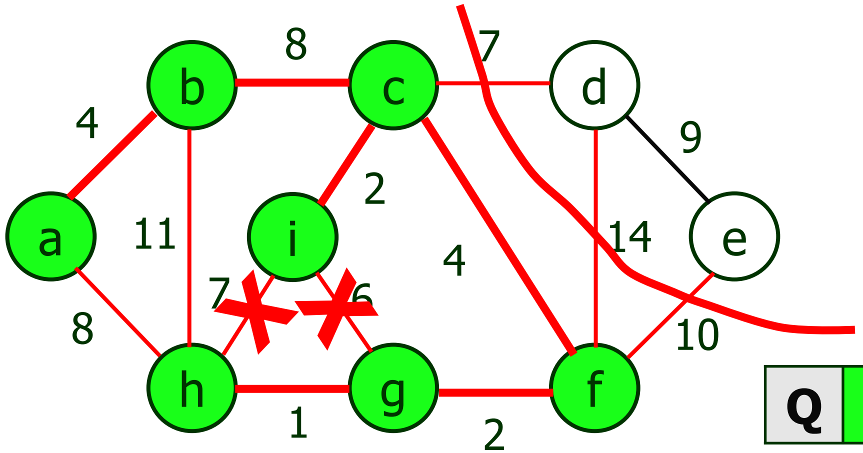
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	10	4	2	1	2
Π	/	a	b	c	f	c	f	h	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

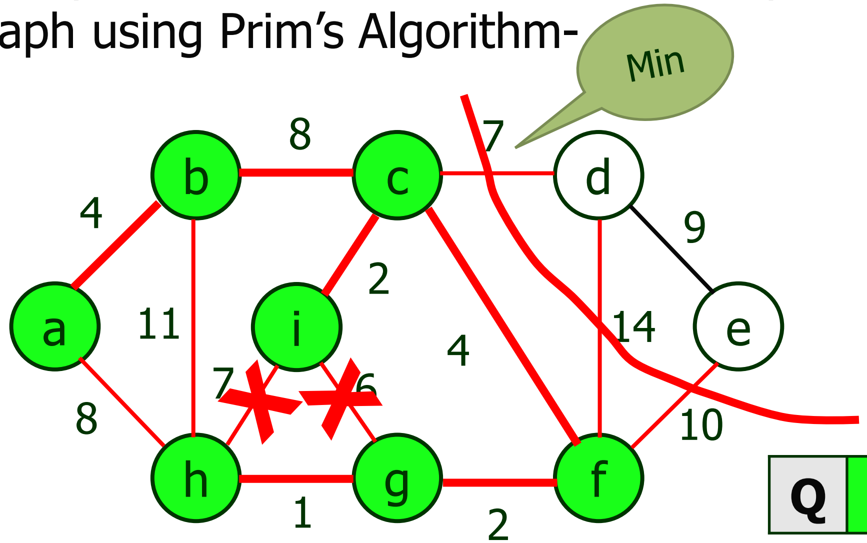
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	10	4	2	1	2
Π	/	a	b	c	f	c	f	h	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

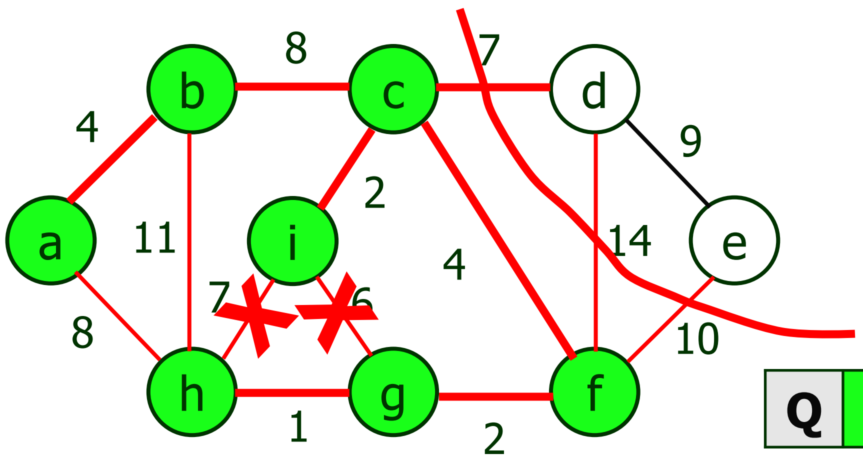
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	10	4	2	1	2
Π	/	a	b	c	f	c	f	h	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

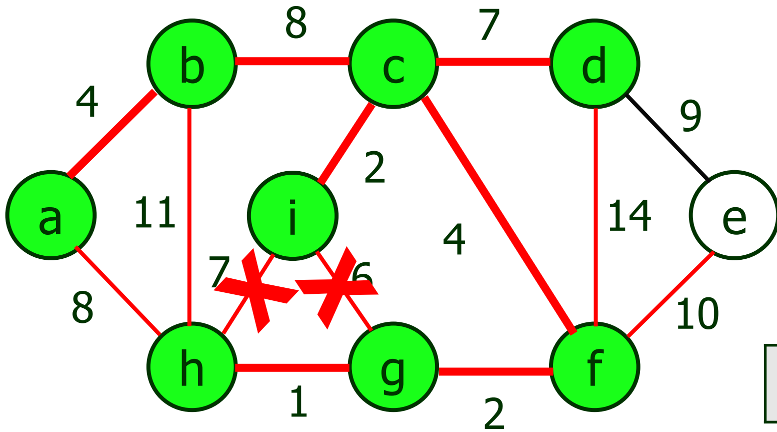
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	10	4	2	1	2
Π	/	a	b	c	f	c	f	h	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

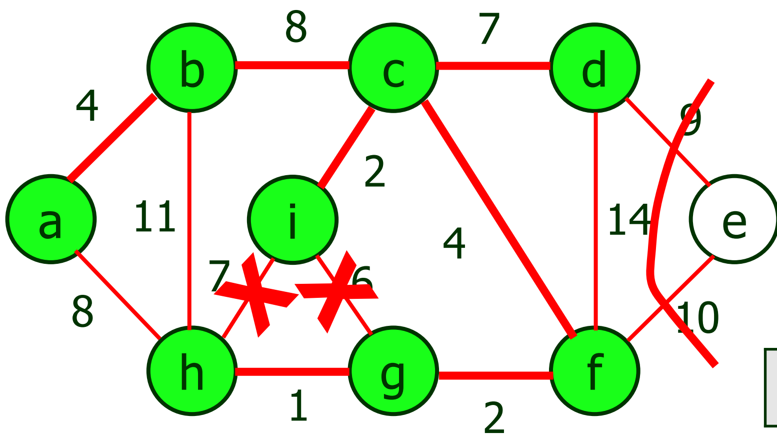
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	10	4	2	1	2
Π	/	a	b	c	f	c	f	h	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

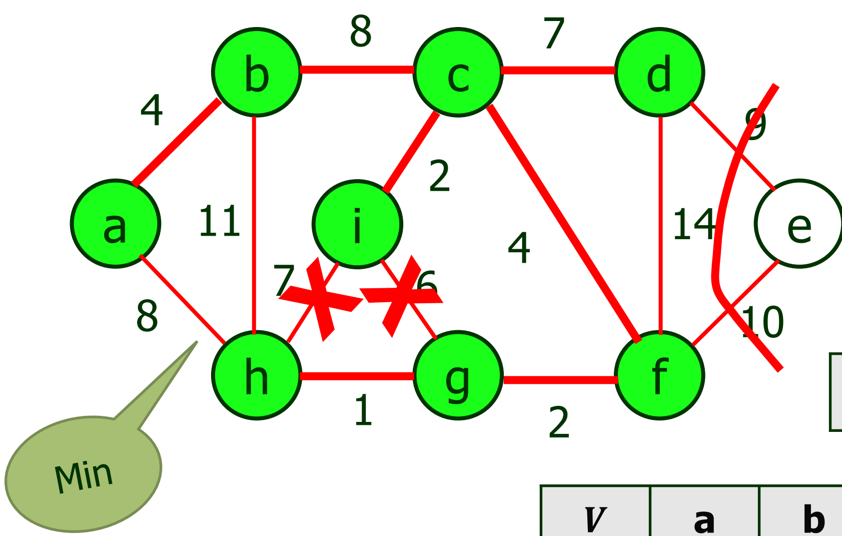
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	9	4	2	1	2
Π	/	a	b	c	d	c	f	h	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

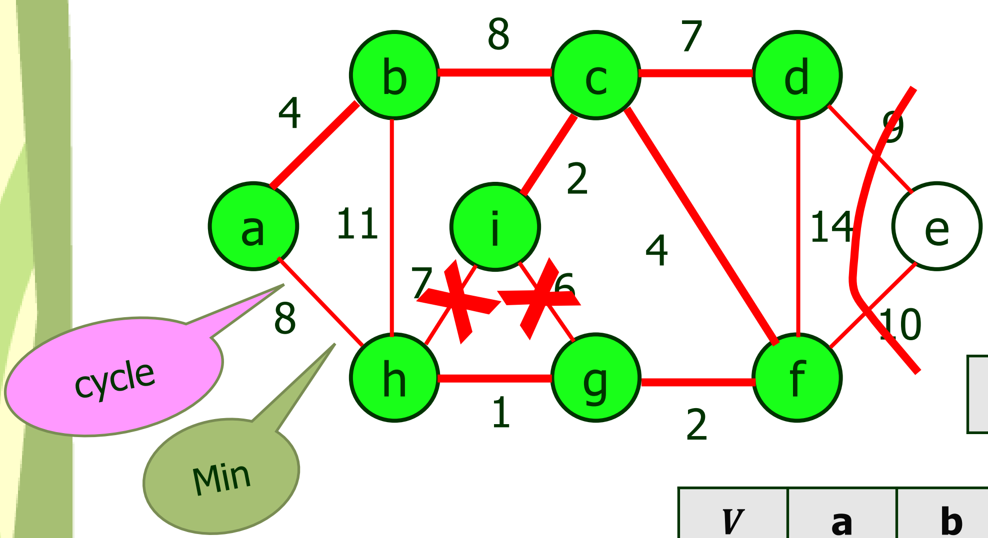
Q	a	b	c	d	e	f	g	h	i

V	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
key	0	4	8	7	9	4	2	1	2
Π	/	a	b	c	d	c	f	h	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

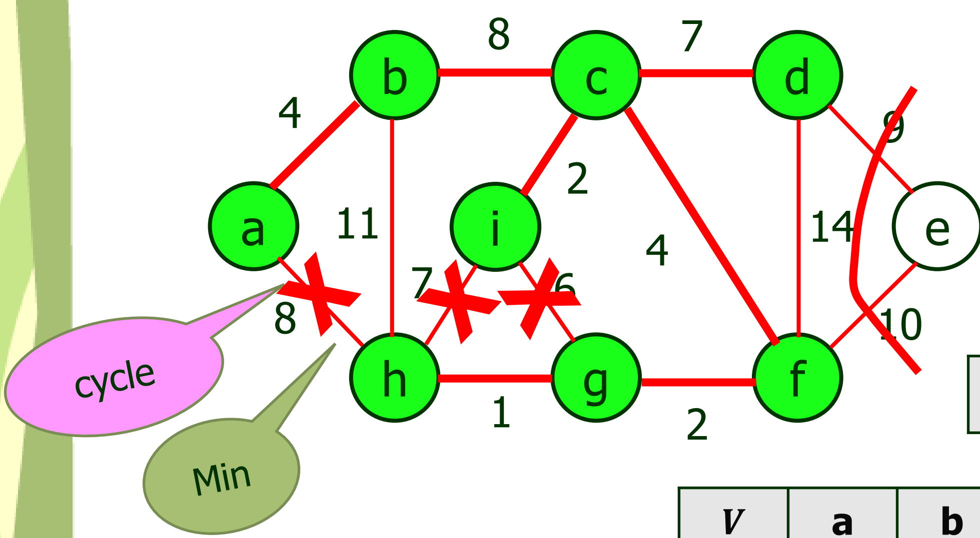
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	9	4	2	1	2
Π	/	a	b	c	d	c	f	h	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

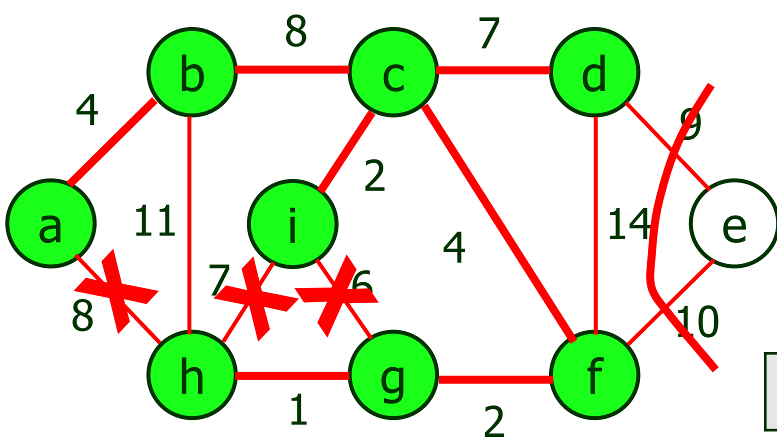
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	9	4	2	1	2
Π	/	a	b	c	d	c	f	h	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

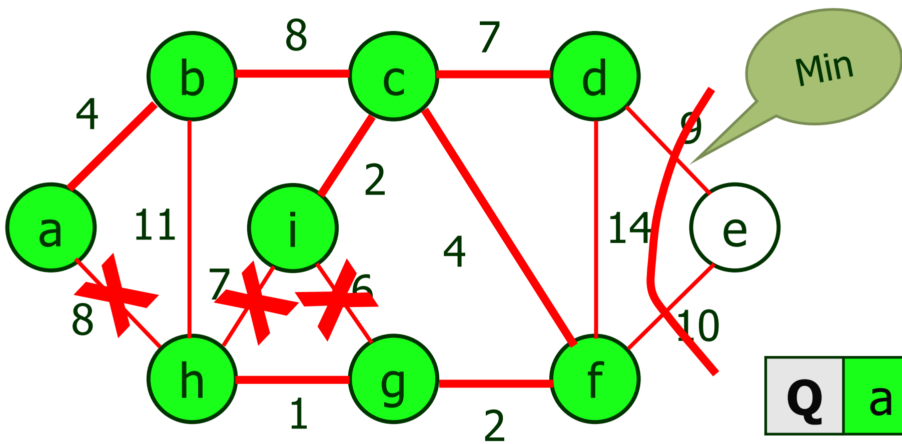
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	9	4	2	1	2
Π	/	a	b	c	d	c	f	h	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

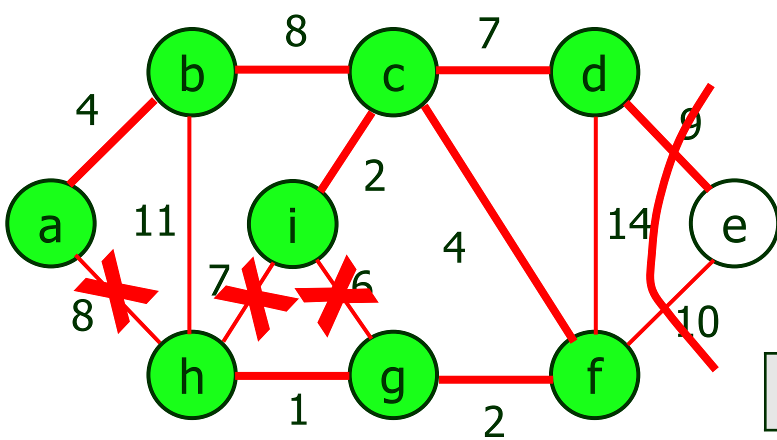
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	9	4	2	1	2
Π	/	a	b	c	d	c	f	h	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

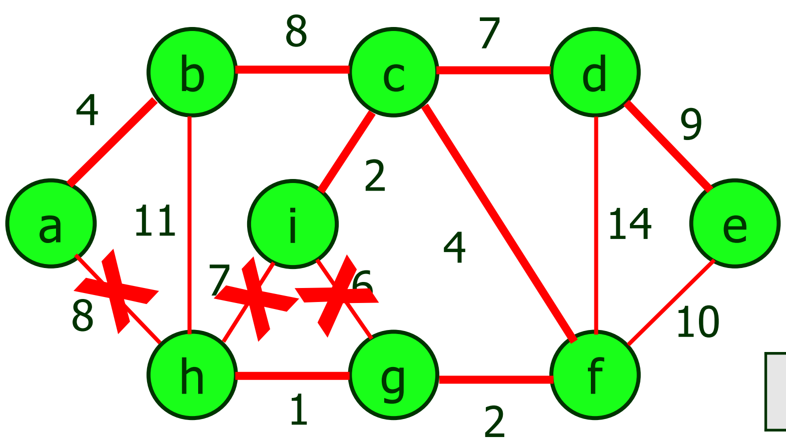
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	9	4	2	1	2
Π	/	a	b	c	d	c	f	h	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

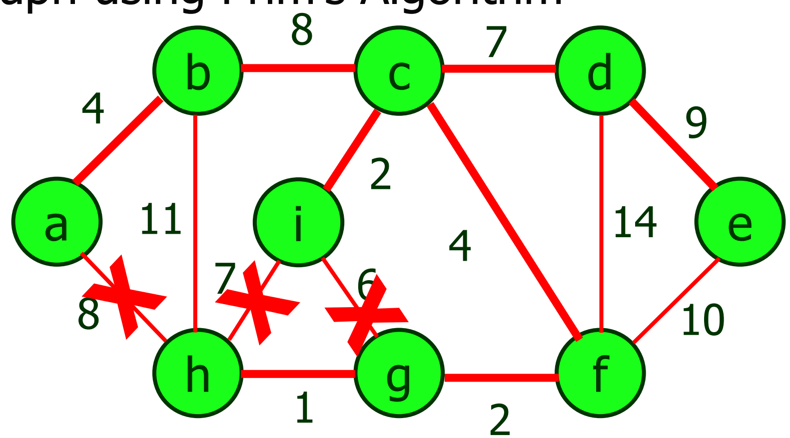
Q	a	b	c	d	e	f	g	h	i

V	a	b	c	d	e	f	g	h	i
key	0	4	8	7	9	4	2	1	2
Π	/	a	b	c	d	c	f	h	c

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Use step 2 and 3 until MST form

Since all the vertices have been included in the MST, so stop the process.

Now, Cost of Minimum Spanning Tree

= Sum of all edge weights

= w[ab]+w[bc]+w[cd]+w[ci]+w[cf]+w[de]+w[gh]+w[fg]

=4+8+7+2+4+9+2+1+=37

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm: (Analysis)**

MST-PRIM(G, w, r)

for each u ∈ V[G]

    do key[u] $\leftarrow \infty$

        π[u] $\leftarrow$ NIL

$O(V)$

key[r] $\leftarrow$ 0 $O(1)$

Q $\leftarrow$ V [G] $O(1)$ by the help of Fibonacci Heap

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm: (Analysis)**

$O(V \lg V)$

```
MST-PRIM(G, w, r)
while Q ≠ Ø
    do u ← EXTRACT-MIN(Q)
        for each v ∈ Adj[u]
            do if v ∈ Q and w(u, v) < key[v]
                then π[v] ← u
                    key[v] ← w(u, v)
```

$O(\lg V)$

$O(V)$

$O(E)$

Decrease Keyof Fibonacci Heap will take $O(1)$

Time Complexity of Prim's Algorithm

=Build min Heap()+Extract Min()+Decrease Key()

$=O(E) + O(V \lg V) + O(E.1)$ {In case of Sparse graph}

$=O(E + V lg\ V)$

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
  - **Prim's Algorithm: (Analysis)**

$O(V \lg V)$

MST-PRIM(G, w, r)

while Q ≠ Ø

do u ← EXTRACT-MIN(Q) $O(\lg V)$

for each v ∈ Adj[u] $O(V)$

do if v ∈ Q and w(u, v) < key[v]

then π[v] ← u $O(E)$

key[v] ← w(u, v)

Decrease Keyof Fibonacci Heap will take $O(1)$

Time Complexity of Prim's Algorithm

=Build min Heap()+Extract Min()+Decrease Key()

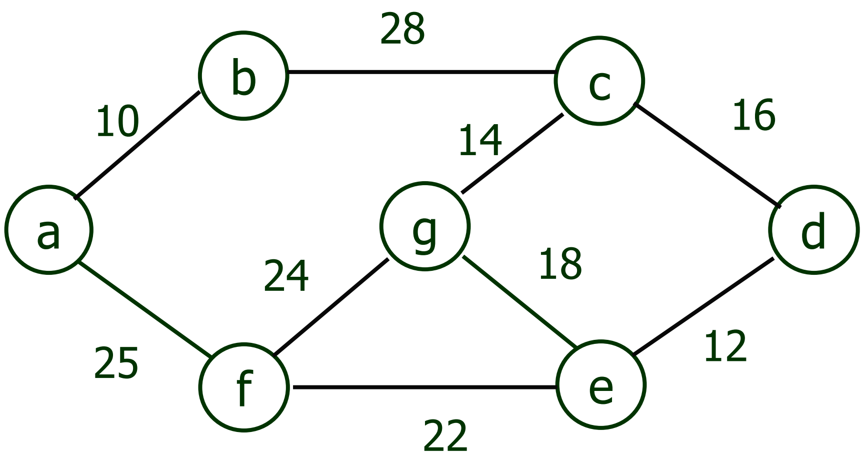
$=O(V) + O(V \lg V) + O(V^2.1)$ {In case of Dense graph ($E = V^2$)}

$=O(V^2 + Vlg\ V) = O(V^2)$

Greedy Algorithm

- **Problem 4: Minimum Spanning Tree problem**
- **Prim's Algorithm:**

Example 1: Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-



Thank u