Data Structures and Algorithms

Prof. Ganesh Ramakrishnan,
Prof. Ajit Diwan,
Prof. D.B. Phatak

Department of Computer Science and Engineering
IIT Bombay

Session: Spanning Tree Algorithm
(Prim's Algorithm)

Prim's Algorithm

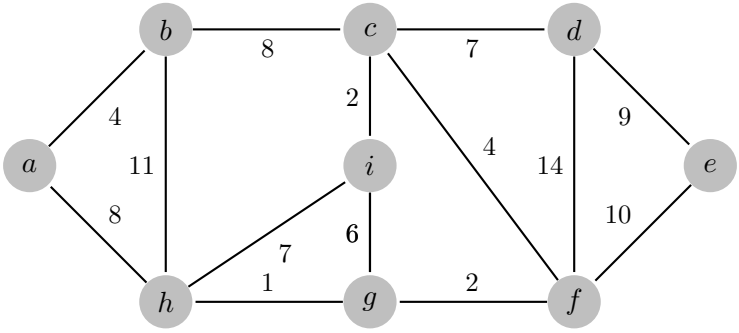
1. Greedy algorithm
2. Finds minimum spanning tree for a weighted undirected graph.

Prim's Algorithm

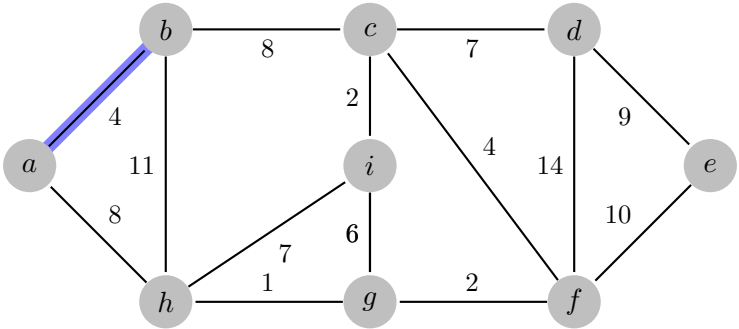
```
Algorithm MST-Prim(G, w, r)
P ← new Min-heap(key)
for u ∈ G.getVertices() do
    key[u] = ∞
    predecessor[u] = NULL
    P.insert(u)
end for
key[r] = 0
while P.isNotEmpty() do
    u ← P.getMin()
    for v ∈ G.adjacentVertex(u) do
        if v ∈ P & w(u, v) < key(v) then
            predecessor[v] = u
            key[v] = w(u, v)
        end if
    end for
end while
```

Figure: Prim's Algorithm

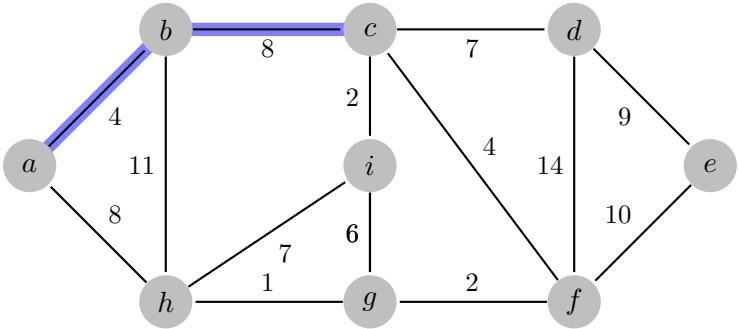
Prim's Algorithm



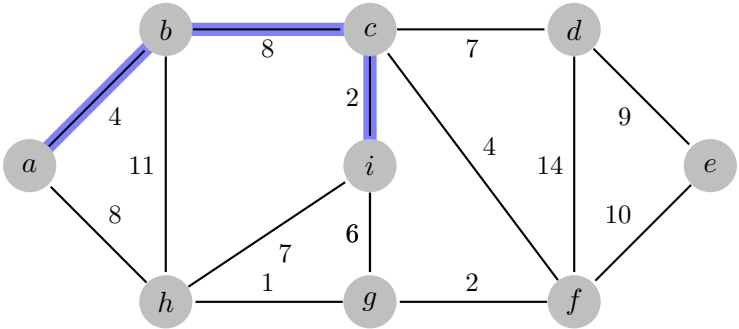
Prim's Algorithm



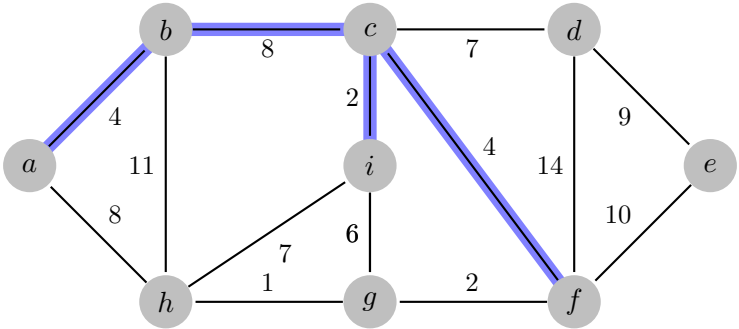
Prim's Algorithm



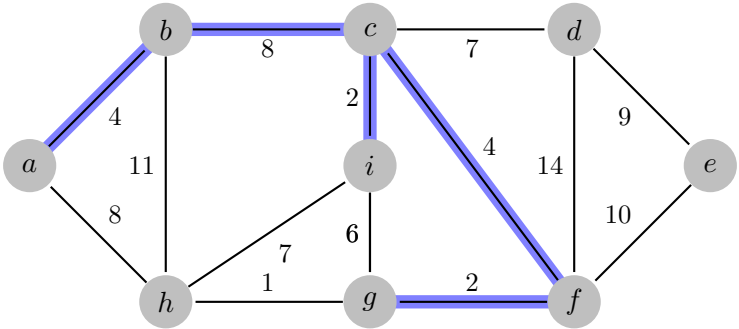
Prim's Algorithm



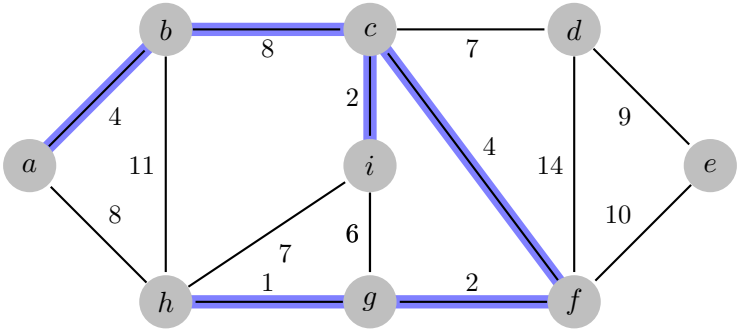
Prim's Algorithm



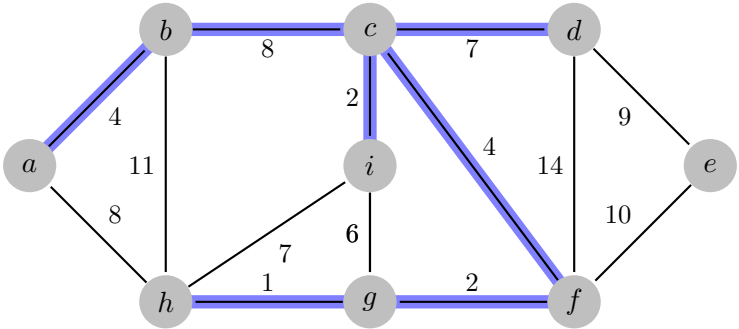
Prim's Algorithm



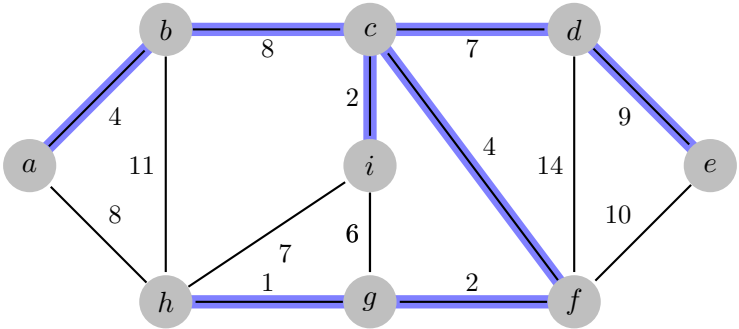
Prim's Algorithm



Prim's Algorithm



Prim's Algorithm



Analysis of Prim's Algorithm

Algorithm MST-Prim(G, w, r) (G)
$P \leftarrow$ new Min-heap(key)
for $u \in G.getVertices()$ **do**
  $key[u] = \infty$
  $predecessor[u] = NULL$
  $P.insert(u) \implies c_1$
end for $\implies c_2 \times |V|$ times
$key[r] = 0$
while $P.isNotEmpty()$ **do**
  $u \leftarrow P.getMin() \implies c_3 \times log|V|$ times
  for $v \in G.adjacentVertex(u)$ **do**
    if $v \in P$ & $w(u, v) < key(v)$ **then**
      $predecessor[v] = u$
      $key[v] = w(u, v) \implies c_4 \times log|V|$ times
    end if
  end for $\implies c_5 \times deg(V)$ times
end while $\implies c_6 \times |V|$ times

Figure: Prim's Algorithm

$$T(n) = c_4c_5 \log |V| \sum_{v \in V} deg(v) + c_3c_6|V| \log |V| + c_1c_2|V| = O((|E| + |V|) \log |V|$$

Thank you