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
(Kruskal's Algorithm)
Content largely adapted from CLRS, Third Edition

Kruskal's Algorithm: Introduction

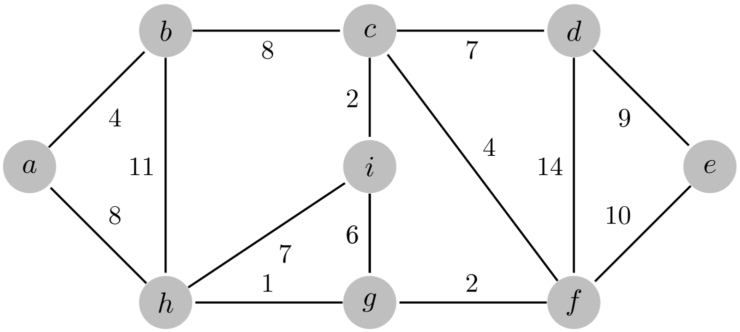
1. Minimum-spanning-tree algorithm using greedy approach
2. Pick the smallest weight edge that does not cause a cycle in the minimum spanning tree
3. Finds an edge of the least possible weight that connects any two sub-trees in the forest
4. It finds a minimum spanning tree by adding increasing cost at each step

Kruskal's Algorithm

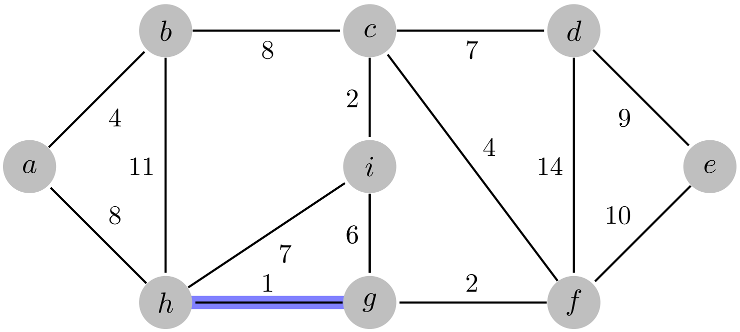
Algorithm MST-Kruskal(G, w)
$\mathcal{T} = \phi$
$SSets[] =$ List of $G.numVertices()$ sequences
$SetID[] =$ List of $G.numVertices()$ integers
for $v \in G.getVertices()$ **do**
    $SSets[v].insert(v)$
    $SetID[v] = v$
end for
SE[] = Sorted edges of G.edges into non decreasing order by weight w
for $edge(u, v) \in SE[]$ **do**
    if $SetId[u] \neq SetId[v]$ **then**
        $\mathcal{T} = \mathcal{T} \cup (u, v)$
        $merge(SSets[u], SSets[v])$
        $SSets[v].empty()$
        $SetId[v] = u$
    end if
end for
return $\mathcal{T}$

Figure: Kruskal's Algorithm

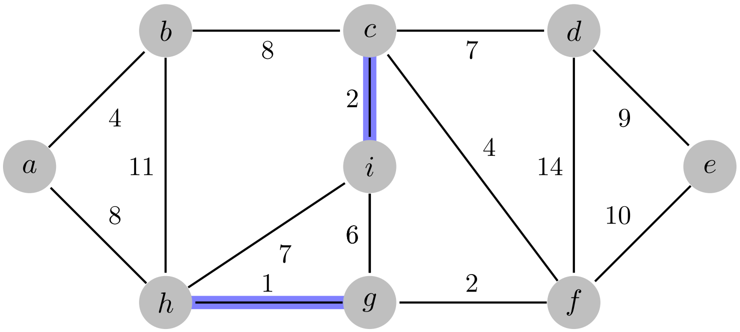
Kruskal's Algorithm



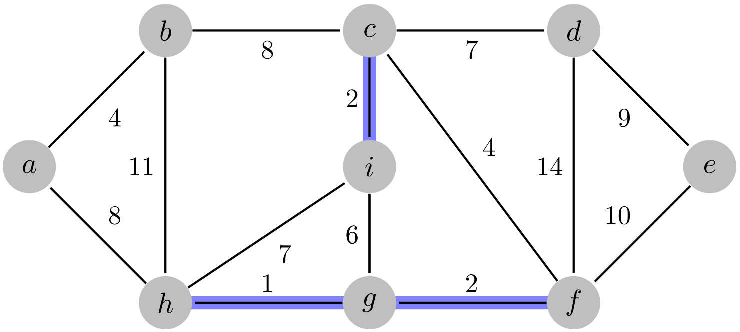
Kruskal's Algorithm



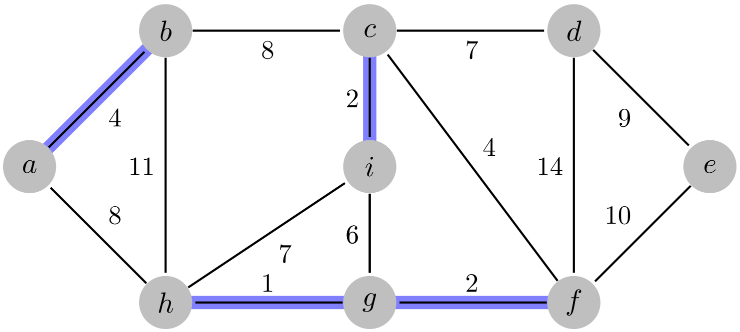
Kruskal's Algorithm



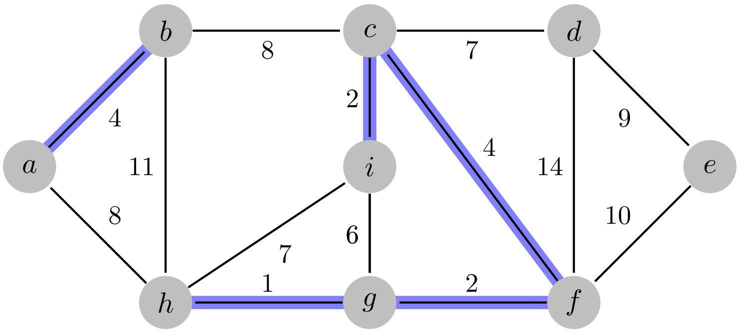
Kruskal's Algorithm



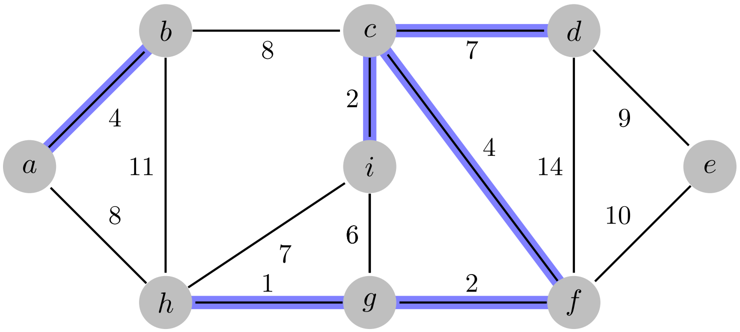
Kruskal's Algorithm



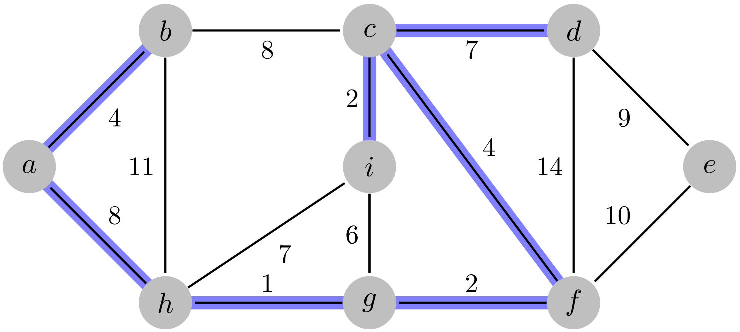
Kruskal's Algorithm



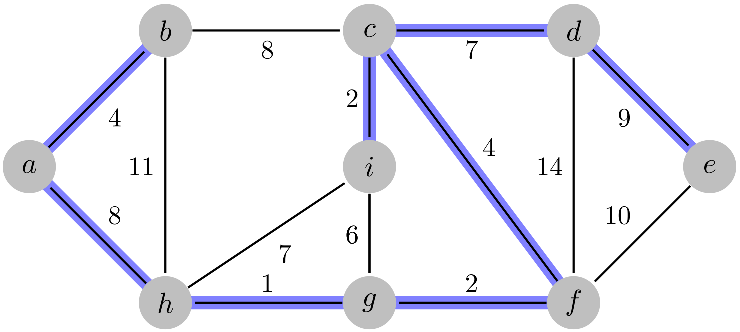
Kruskal's Algorithm



Kruskal's Algorithm



Kruskal's Algorithm



Analysis of Kruskal's Algorithm

```
Algorithm MST-Kruskal(G, w)
T = φ
SSets[] = List of G.numVertices() sequences
SetID[] = List of G.numVertices() integers
for v ∈ G.getVertices() do
    SSets[v].insert(v)
    SetID[v] = v
end for  ⟹  c1  × |V| times
SE[] = Sorted edges of G.edges into non decreasing order by weight w  ⟹  c2  × O|E|log|E| times
for edge(u, v) ∈ SE[] do
    if SetId[u] ≠ SetId[v] then
        T = T ∪ (u, v)  ⟹  c3  × 2|E| times
        merge(SSets[u], SSets[v])  ⟹  c4  × |E| times
        SSets[v].empty()  ⟹  c5  × |E| times
        SetId[v] = u  ⟹  c6  × |E| times
    end if
end for
return T
```

Figure: Kruskal's Algorithm

$T(n) = c_1|V| + c_2|E|log|E| + 2c_3(c_4 + c_5 + c_6)|E| = O(|E|log|E|)$ or $O(|E|log|V|)$

Thank you