Minimum spanning tree
BARUVKA ALGORITHM

Baruvka's Algorithm

- Input is a connected, weighted and un-directed graph.
- Initialize all vertices as individual components (or sets).
- While there are more than one components, do following for each component.
  - Find the closest weight edge that connects this component to any other component.
  - Add this closest edge to MST if not already added.

MST using Baruvka's Algorithm

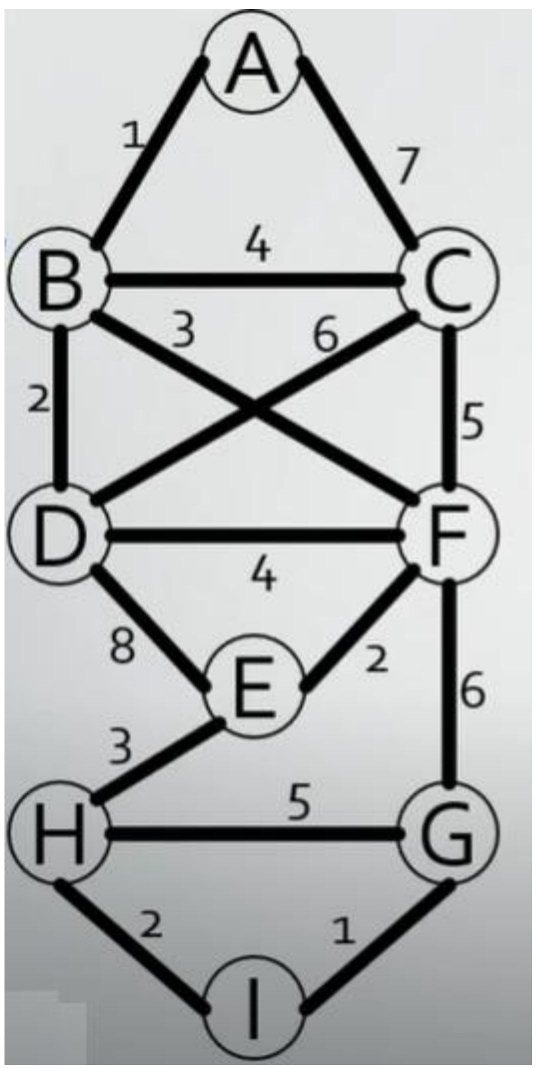
Algorithm BarůvkaMST(G):

Input: A weighted connected graph $G = (V, E)$ with n vertices and m edges
Output: A minimum spanning tree T for G.

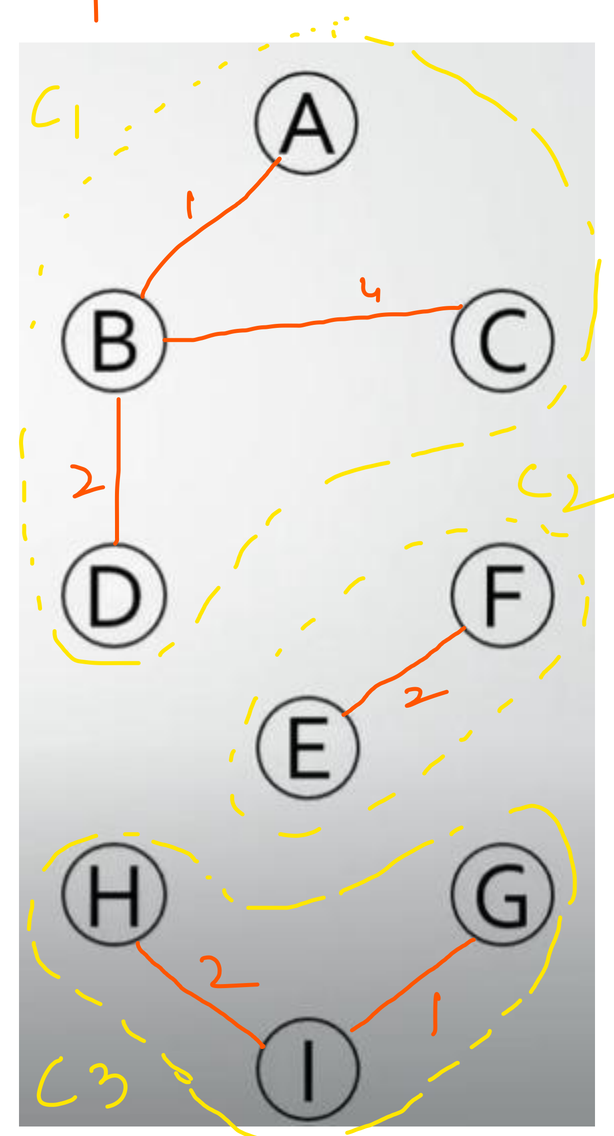
Let T be a subgraph of G initially containing just the vertices in V.
while T has fewer than $n - 1$ edges $\{T$ is not yet an MST$\}$ **do**
  for each connected component C_i of T **do**
    {Perform the MST edge addition procedure for cluster C_i}
    Find the smallest-weight edge $e = (v, u)$, in E with $v \in C_i$ and $u \notin C_i$.
    Add e to T (unless e is already in T).
return T

MST using Baruvka’s Algorithm

G



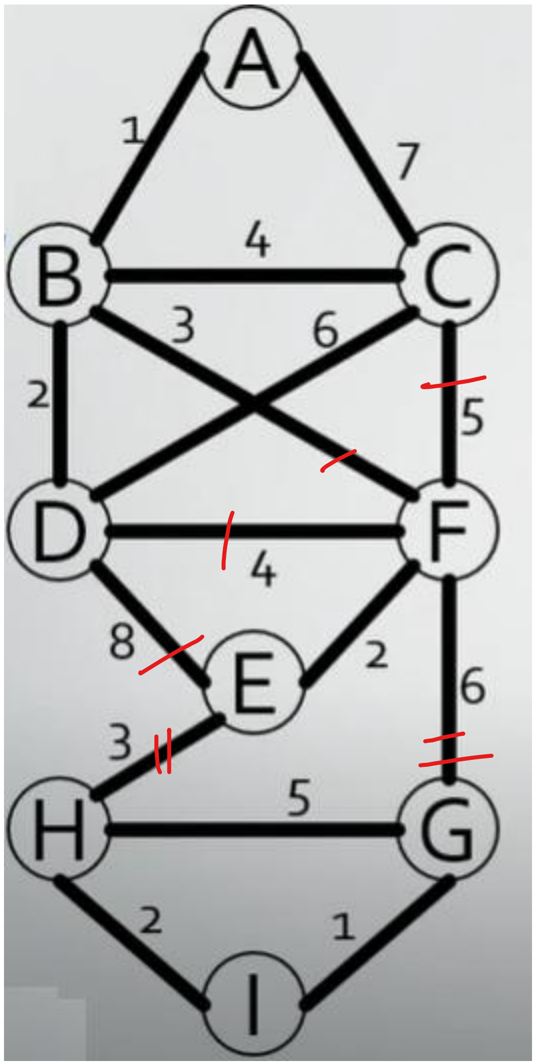
T



{A} - A – B
{B} A – B
{C} B – C
{D} B – D
{E} E – F
{F} E – F
{G} G – I
{H} H – I
{I} I – G

MST using Baruvka's Algorithm

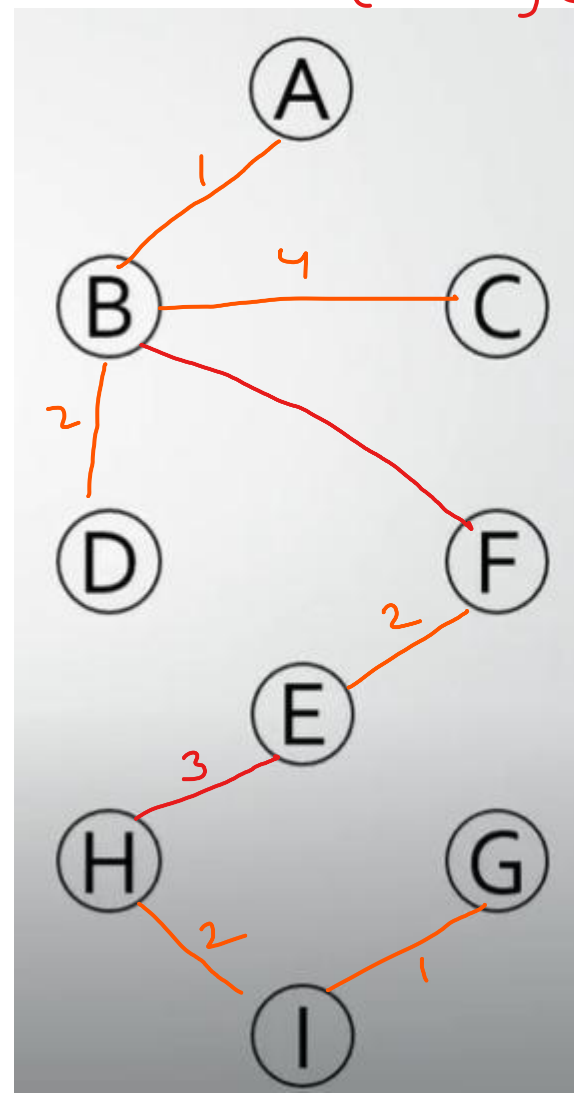
T (n-1) edge



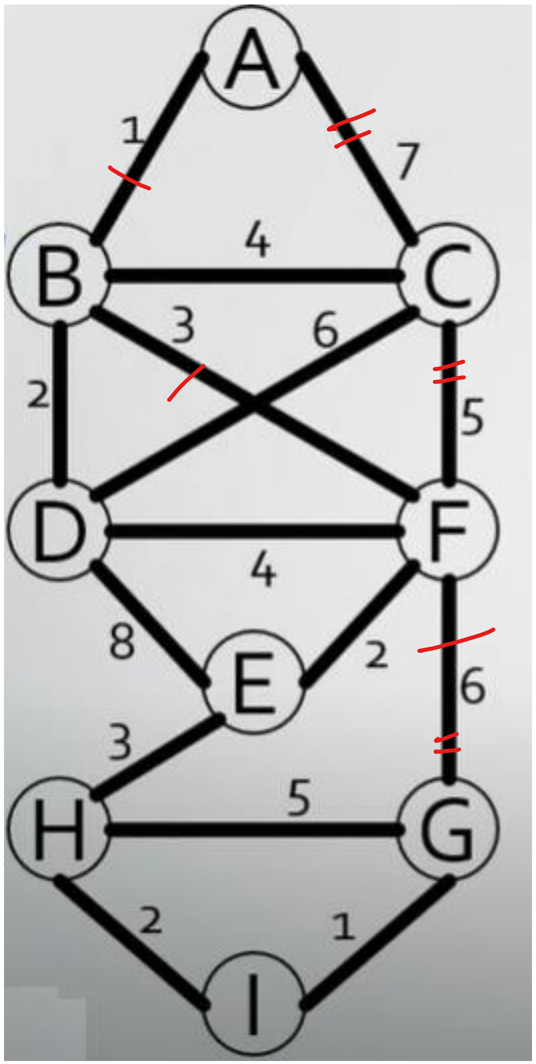
{a, b, c, d} = B-F

{e, f} = B-F

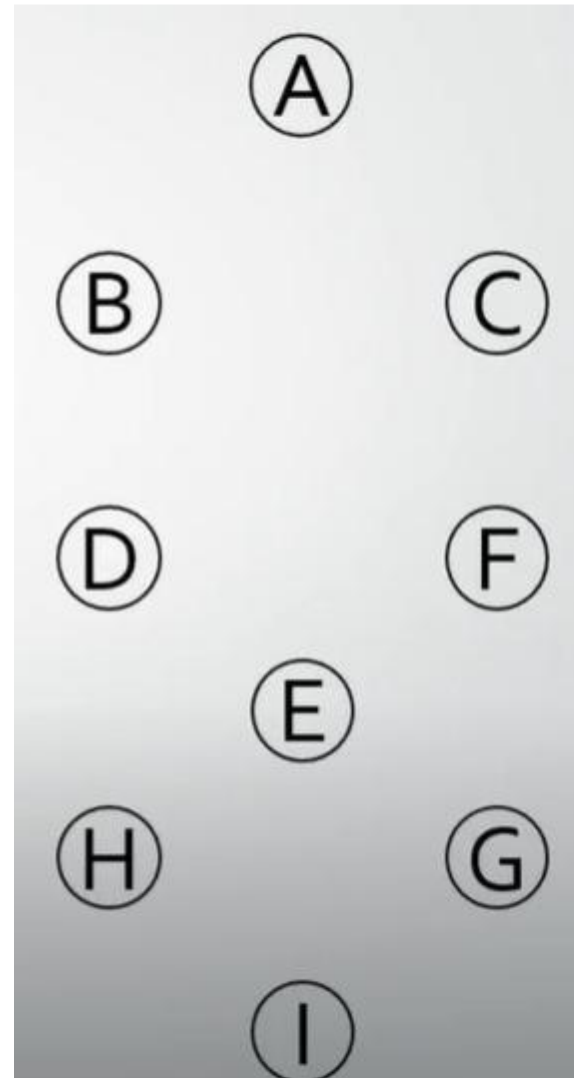
{g, h, i} = E-H



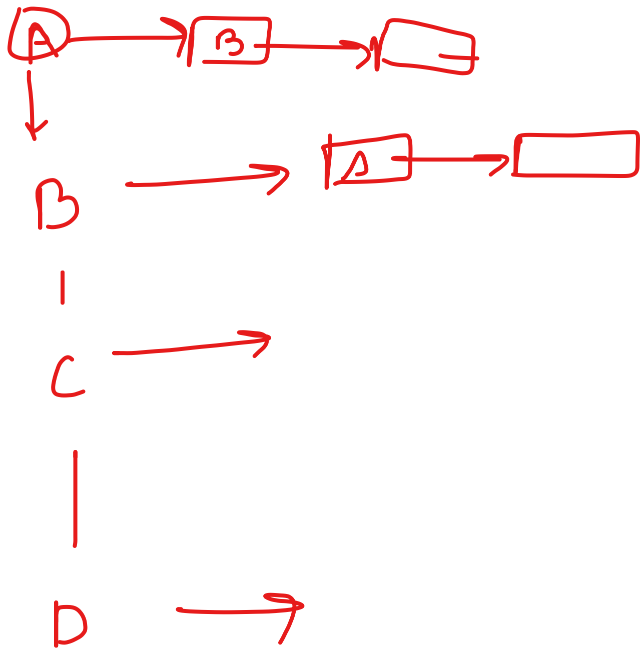
MST using Baruvka's Algorithm



1r
1, 3, 6
7, 5, 6



Adj list E/m edges

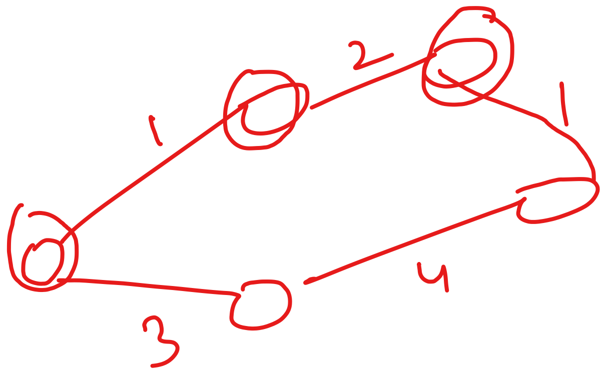


$(O(d))$

① → Each edge is visited $\equiv 2m \equiv O(m)$

② → Iteration $\equiv \log(n)$

Complexity = $O(m \log n)$


1
2
1
3
4