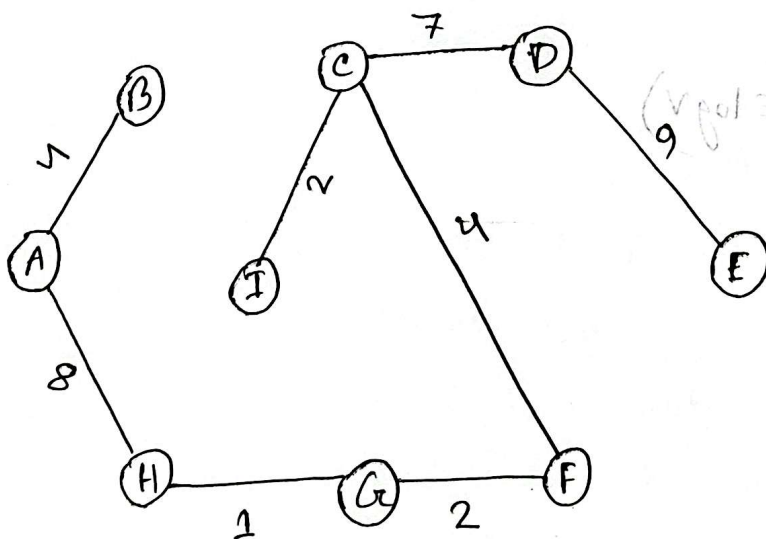
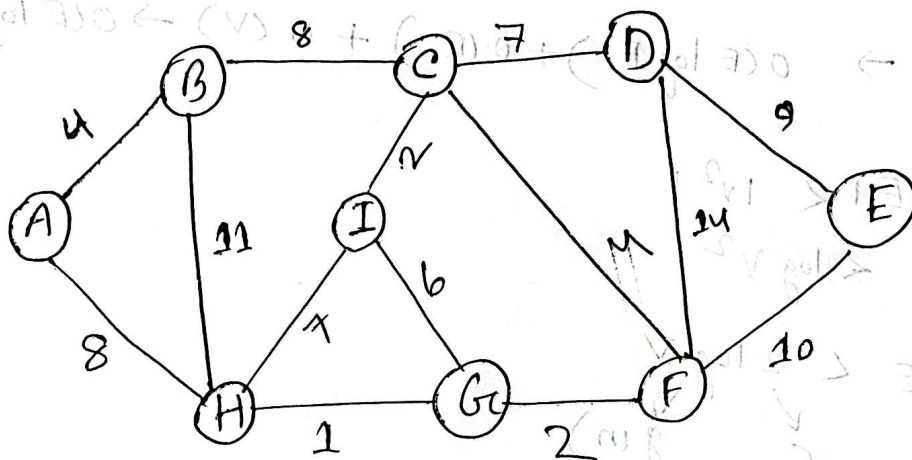


Kruskal Algorithm: Minimum weight spanning tree (જાડા રેખા)

Ascending order ના સ્તરોમાં રજૂ કરવામાં આવે છે. Then, same edge ના ચકાસણી કરવામાં આવે છે કે cycle રચે છે કે નહીં. (જો cycle રચે છે તો તે edge નામત કરવામાં આવે છે)

Simulation:



Edge	Weight
G-H	1 ✓
G-F	2 ✓
C-I	2 ✓
A-B	4 ✓
C-F	4 ✗
G-I	6 ✗ [cycle]
C-D	7 ✓
H-I	7 ✗ [cycle]
A-H	8 ✓
B-C	8 ✗ [cycle]
D-E	9 ✓
E-F	10 ✗
B-H	11 ✗
D-F	14 ✗

Found MST with Total weight = $4 + 8 + 1 + 2 + 4 + 2 + 7 + 9 = 37$

Time Complexity - Kruskal's Algorithm

Sorting E edges $\rightarrow O(E \log E)$

Selecting E edges $\rightarrow O(E)$

Checking V nodes if is in the same component (Disjoint sets) $\rightarrow O(V)$

Overall $\rightarrow O(E \log E) + O(E) + O(V) \rightarrow O(E \log E)$

As, $|E| < |V|^2$

$\Rightarrow \log E < \log V^2$

$\Rightarrow \log E < 2 \log V$

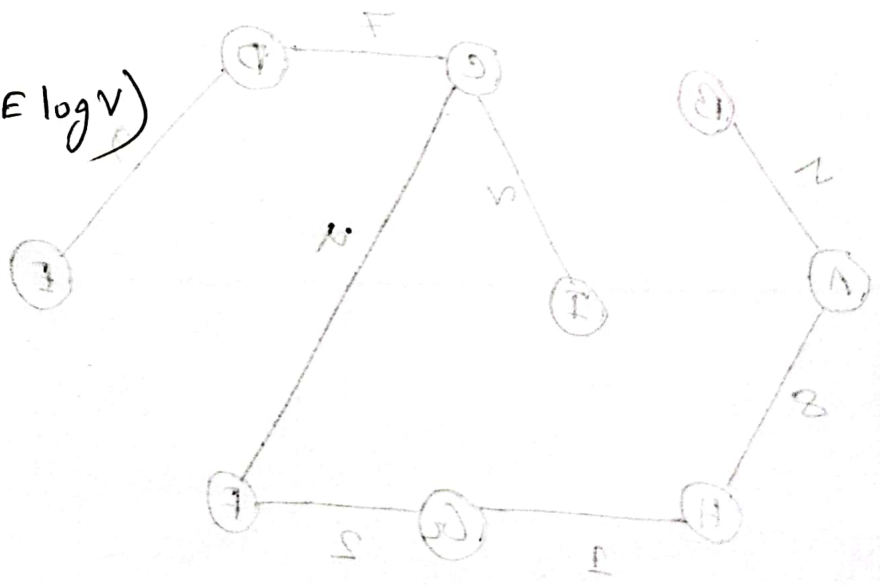
\downarrow
 $f(n)$

\downarrow
 c

\downarrow
 $g(n)$

$\log E = O(\log V)$

$O(E \log E) = O(E \log V)$



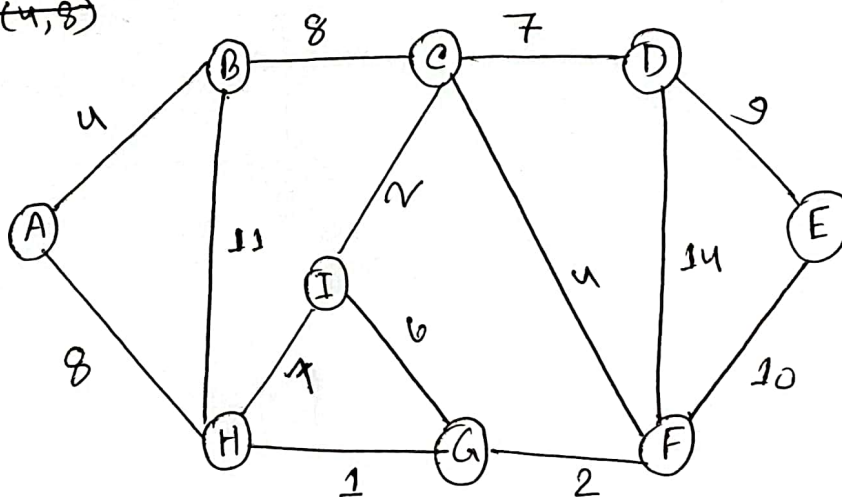
Total weight = 2+8+1+7+4+3+5+2+1+3+2 = 38

Prim's Algorithm —

Source node u select, Then minimum weight A_u ,
~~First~~ First a source node add u , then
 adjacent nodes (edges) v and u way select minimum A_v
 select A_v .

Simulation —

Adding A to MST
 $\min(u, 8)$



Steps \Rightarrow 1) Add A to MST.

(2) $\min(4, 8)$, Add B, A-B to MST

(3) $\min(8, 11, 8)$, Add C, B-C to MST

(4) $\min(8, 11, 2, 4, 7)$, Add I, C-I to MST.

(5) $\min(8, 11, 4, 7, 6, 7)$, Add F, C-F to MST.

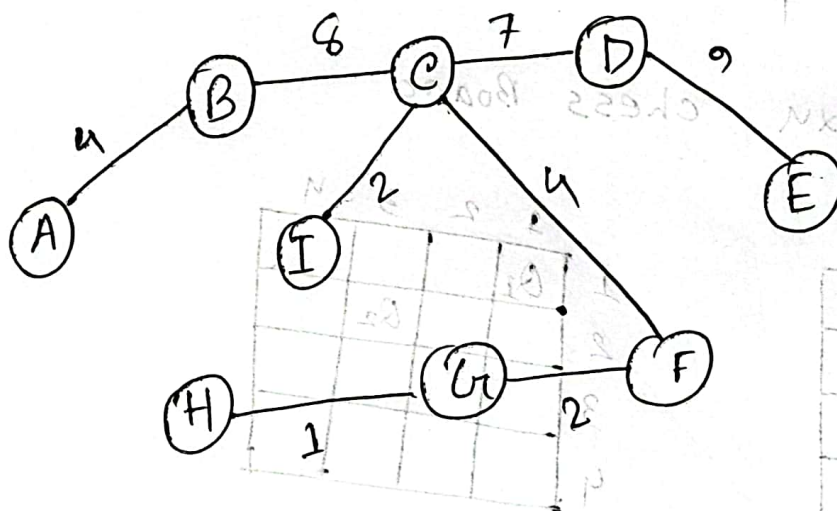
(6) $\min(8, 11, 7, 6, 7, 2, 10, 14)$, Add G, F-G to MST.

(7) $\min(8, 11, 7, 7, 10, 14)$, Add H, G-H to MST.

(8) $\min(7, 10, 14)$, Add D, C-D to MST.

(g) $\text{Min}(9, 10)$, Add E, D-E to MST.

Found MST with total weight = $4 + 8 + 2 + 4 + 2$
 $= 37$



Complexity \Rightarrow

Using Adjacency Matrix - $O(V^2)$

Using Adjacency List - $O(E \log V)$

Backtracking - All possible solutions

N QUEEN Problem -

same row, same column and Diagonally

Simulation - 4x4 chess Board

	1	2	3	4
1	Q ₁			
2				
3				
4				

	1	2	3	4
1	Q ₁			
2			Q ₂	
3	X	X	X	X
4				

	1	2	3	4
1		Q ₁		
2				
3				
4				

	1	2	3	4
1		Q ₁		
2				Q ₂
3	Q ₃			
4				

Q₁ Q₂
2 4

	1	2	3	4
1	Q ₁			
2			Q ₂	
3				
4				

	1	2	3	4
1	Q ₁			
2				Q ₂
3			Q ₃	
4	X	X	X	X

Q₃ Q₄
1 3

	1	2	3	4
1	Q ₁			
2				Q ₂
3				
4				

	1	2	3	4
1		Q ₁		
2				Q ₂
3	Q ₃			
4			Q ₄	

	1	2	3	4
1			Q ₁	
2				
3				
4				

	1	2	3	4
1			Q ₁	
2	Q ₂			
3				
4				

	1	2	3	4
1			Q ₁	
2	Q ₂			
3				Q ₃
4				

	1	2	3	4
1			Q ₁	
2	Q ₂			
3				Q ₃
4		Q ₄		

Q₁
3

Q₂
1

Q₃
4

Q₄
2

Back tracking

$$T(n) = n!$$

complexity -

Because 1st queen at column n.
2nd queen at (n-1), then 3rd
queen (n-2) and so on

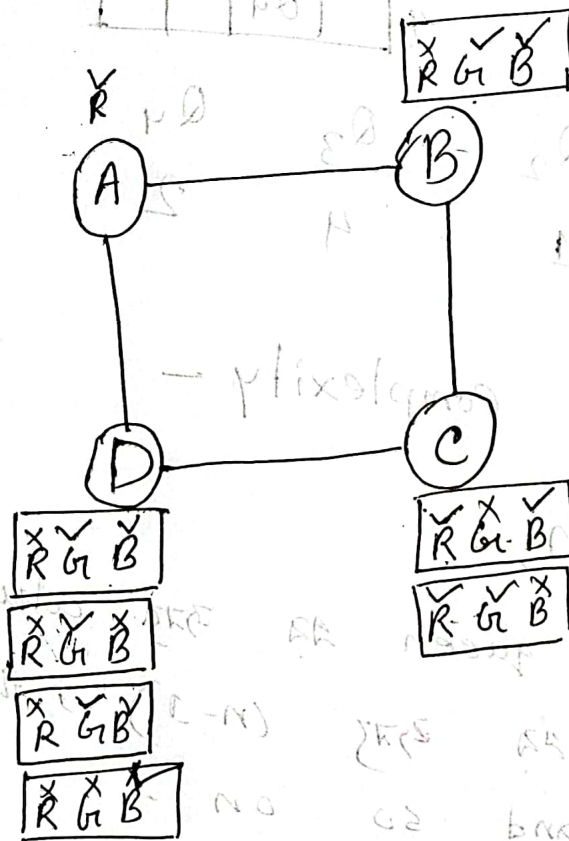
Graph Colouring —

Adjacent node same

complete 2CN Box

Simulation \Rightarrow

$k = 3$ (RGB)



Combinations —

(1) RGRG

(4) RBRG

(2) RGRB

(5) RB RB

(3) RGRG

(6) RBGB