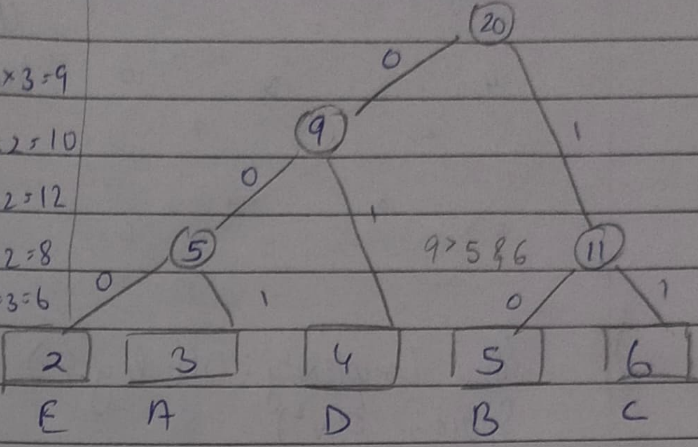
AA Mod 3 Self Learning Topics

- Huffman Coding

Message: BCCABBDDAECCBBAEDDCC

Char	Count	Code	
A	3	001	3×3=9
B	5	10	5×2=10
C	6	11	6×2=12
D	4	01	4×2=8
E	2	000	2×3=6
	20		



add min^m ones together, left = 0 right = 1

4th column total = 45 bits

5 alphabets × 8 bits = 40 bits

Total no. of 0s & 1s in Code = 12 bits

$\sum d_i \times f_i = 3\times2 + 3\times3 + 2\times4 + 2\times5 + 2\times6$

Message: 001 11 11 01 10 11 11 ......

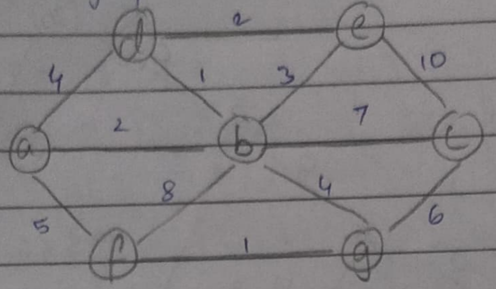
Time Complexity : O(n log n)   Space : O(n)

```
function HuffmanCoding (data):
Compute character frequencies
Create a priority queue (min-heap) Q of characters & their frequencies
while Q has more than one node:
    Combine 2 lowest freq. nodes into a new node, update freq
    Insert new node into Q.
Construct Huffman tree from Q
Generate Huffman codes by traversing Huffman tree.
Return Huffman code
```

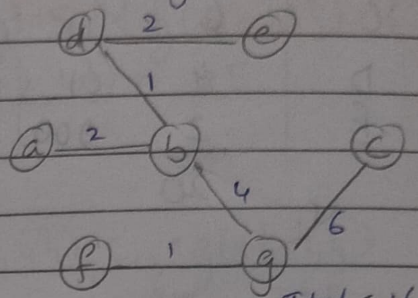
• Prim's Algorithm

Minimum Spanning Tree :-

Subgraph : all vertices, connected, no cycles, edges weight's min



Choose a start vertex,
keep including connected edges
[no cycle]



Total = 16

Time Complexity : O(E log E)
Space : O(V²)

```
Function Prim (graph):
Initialize empty set 'MST' to store MST
Initialize priority queue (min-heap) 'pq' to store edges sorted by wt.
Select an arbitrary vertex 'start' to begin MST construction
Add all edges incident to 'start' to 'pq'
Mark 'start' as visited
while 'pq' is not empty:
        Extract the edge with min weight e from eq
        If both endpoints of e are alr visited, continue to next iteration
        Add e to MST
        Mark endpoint of e that is not visited, as visited
        Add all edges incident to newly visited vertex to pq.
Return the MST
```