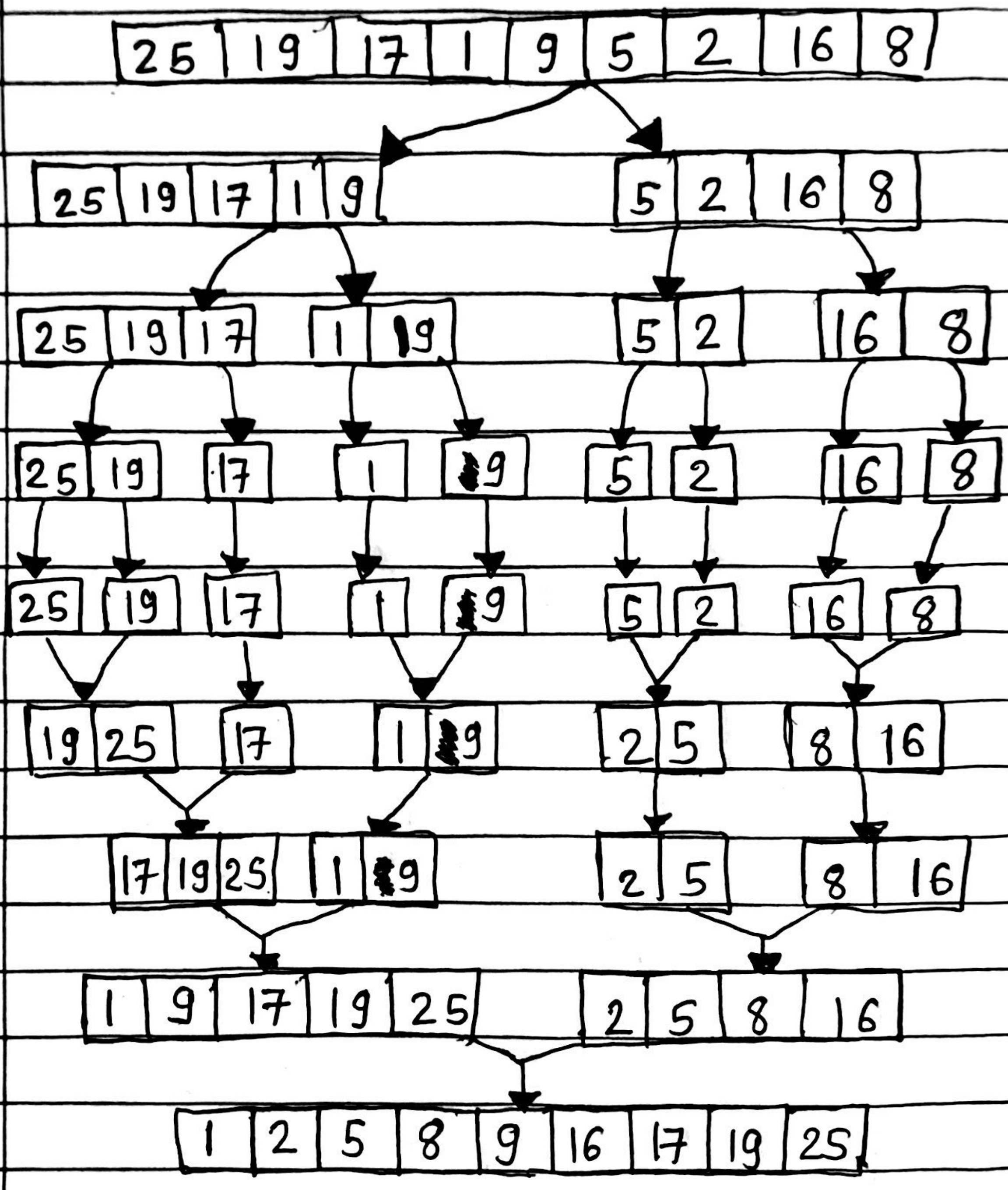
Jenil Pandya 10179217

Q.1 Mergesort Visulization



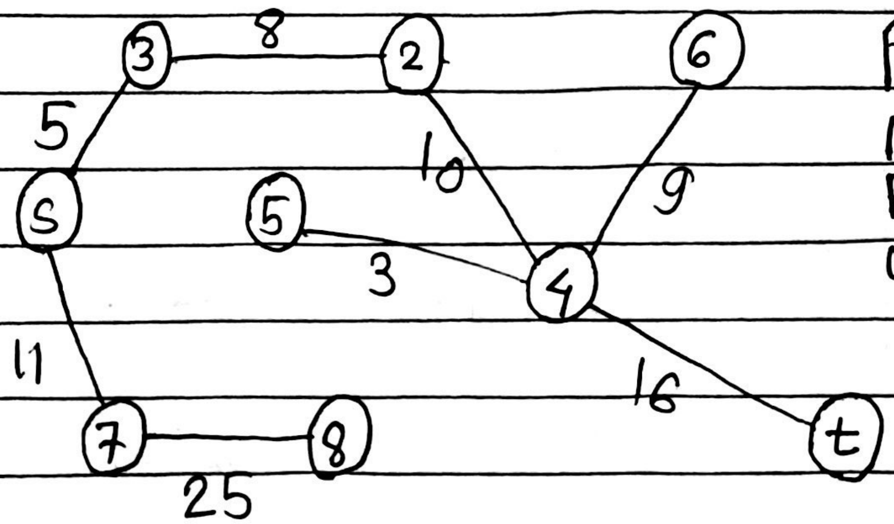

Done sorting!

Pandya
Jenil

# Finding MST using Prim's

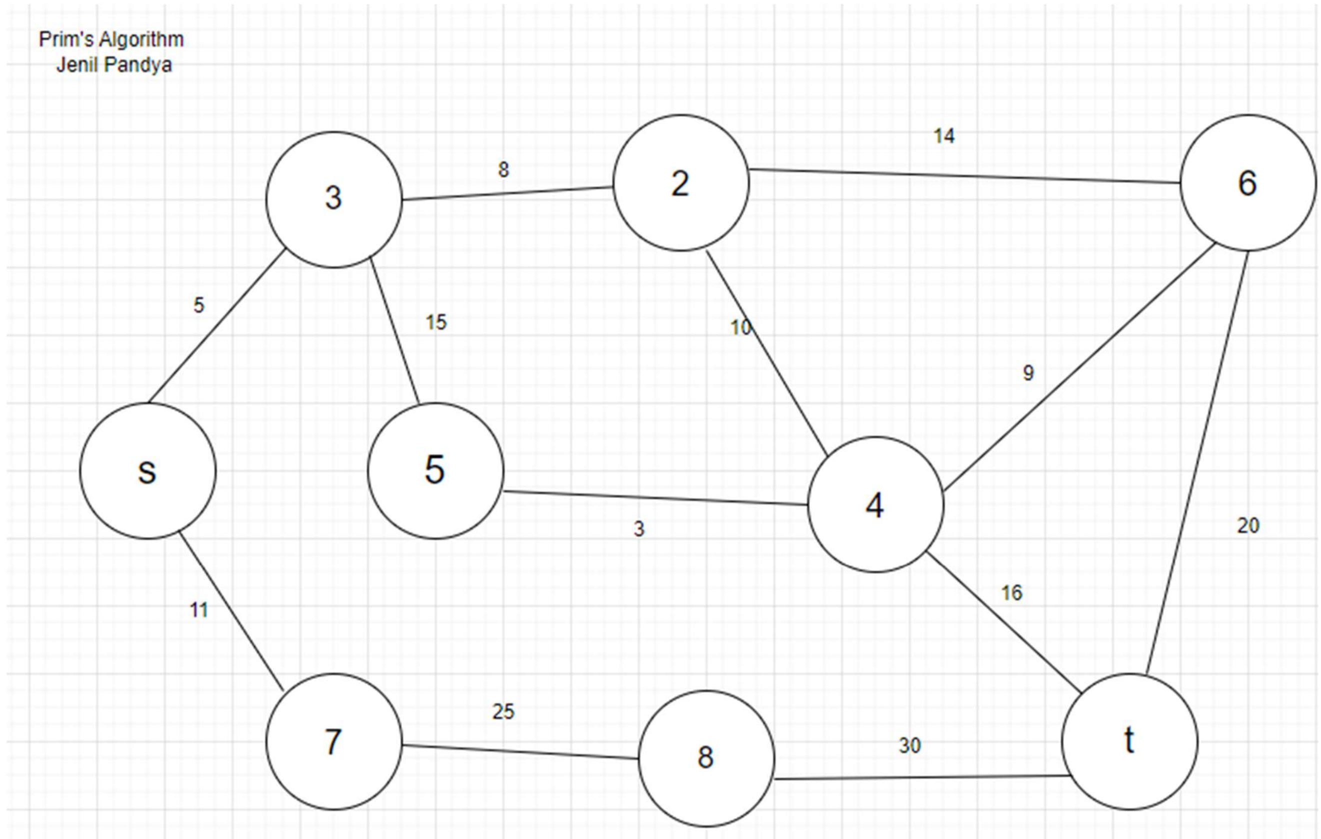
| | Prim's Edges in order | Edge weight | |
|---|---|---|---|
| 1 | 5 → 3 | 5 | |
| 2 | 3 → 2 | 8 | |
| 3 | 2 → 4 | 10 | |
| 4. | 4 → 5 | 3 | |
| 5. | 4 → 6 | 9 | Forming |
| 6. | 5 → 7 | 11 | $2 \xrightarrow{14} 6$ Cycle |
| 7. | 4 → t | 16 | Discarded |
| 8. | 7 → 8 | 25 | $3 \xrightarrow{15} 5$ |
| | Weight total | 87 | $8 \xrightarrow{30} t$ |

Final Graph using Prim's



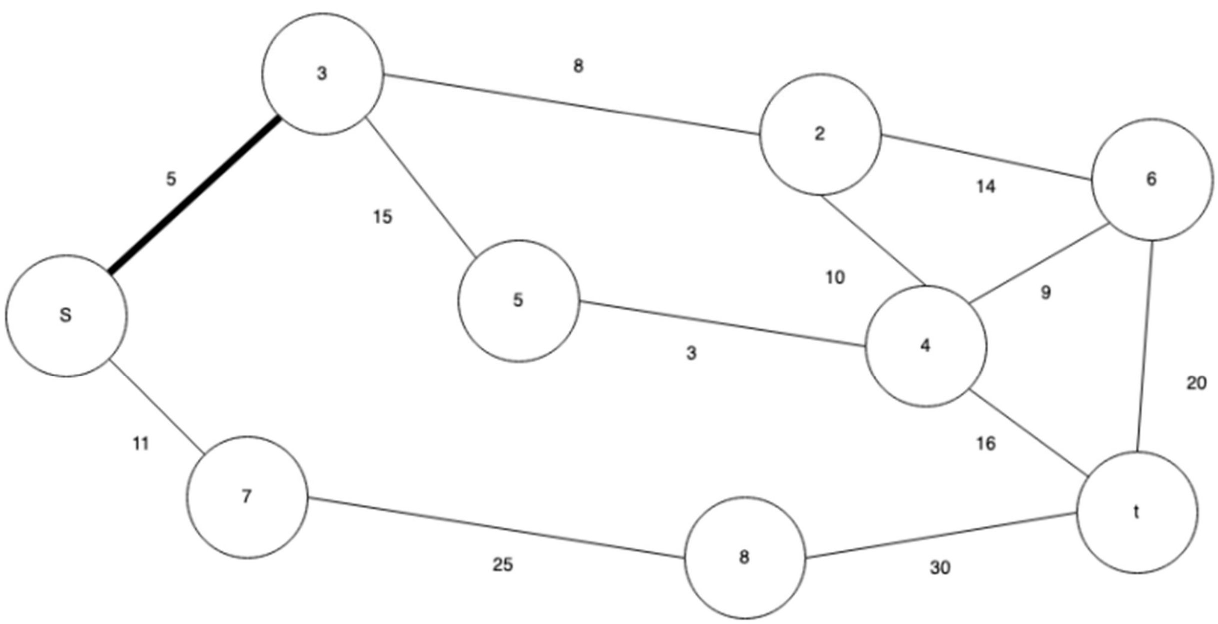

Prim's MST Final weight

$\boxed{87}$



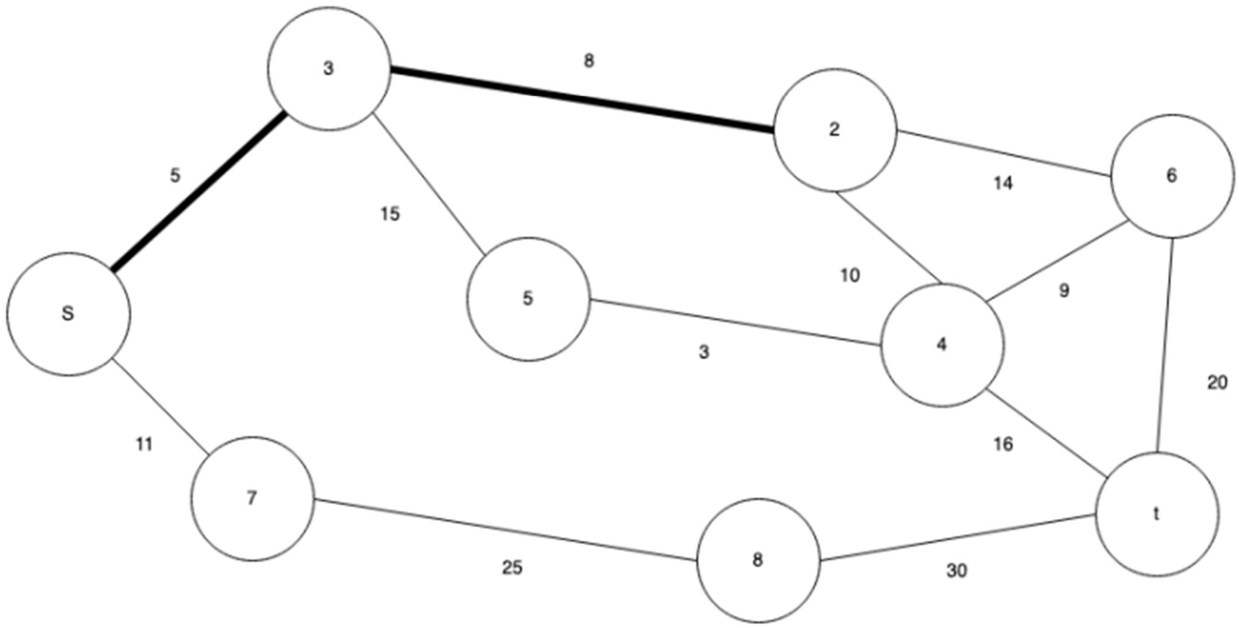

Step 1: -

From node ‘s’ selecting edge with weight 5



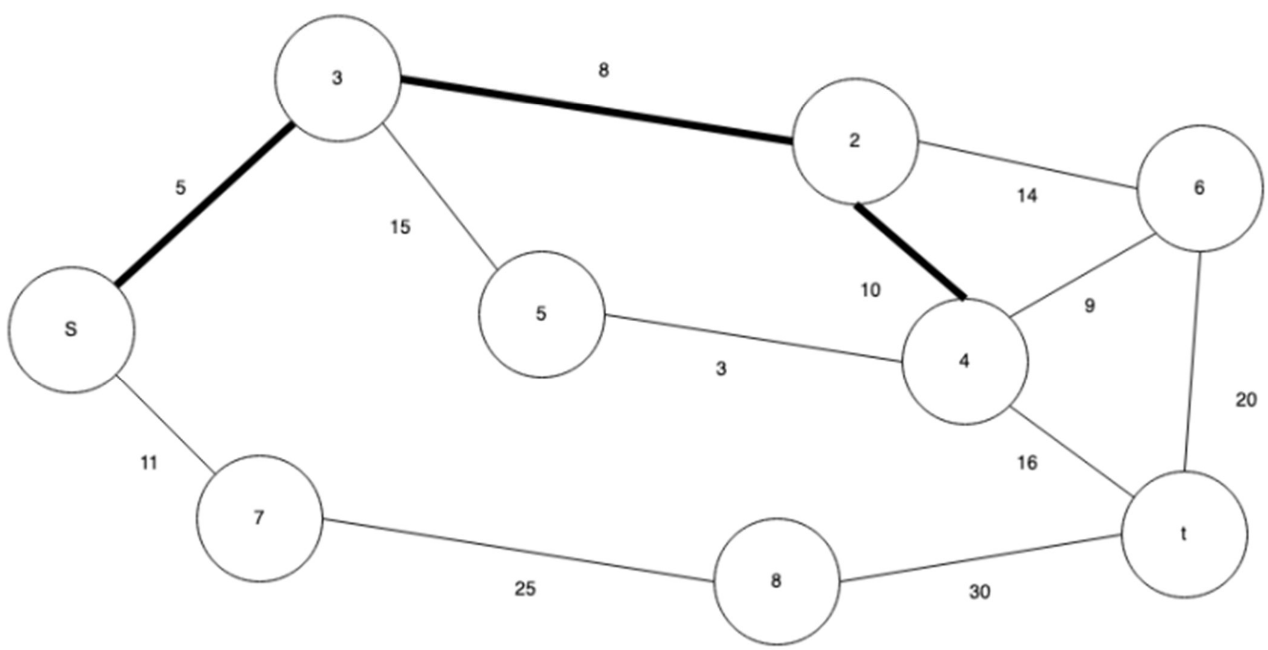
Step 2: -

From node '3' the lowest edge is the one with weight as 8



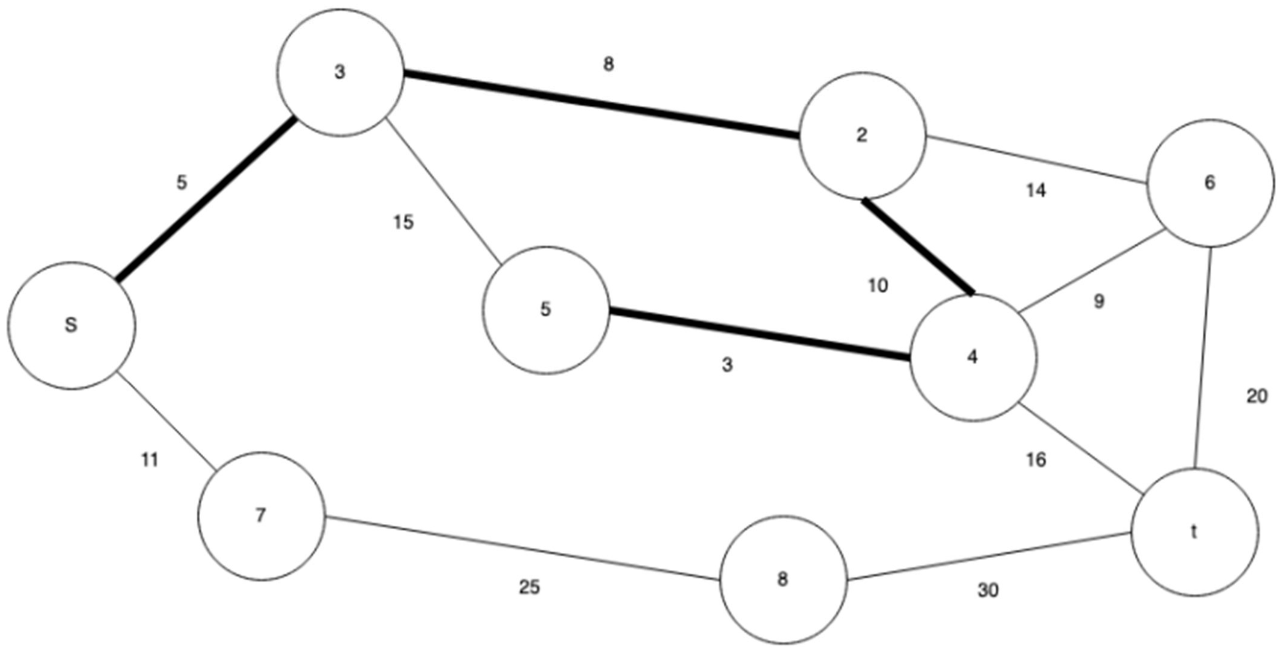

Step 3: -

From node '2' selecting edge with weight 10



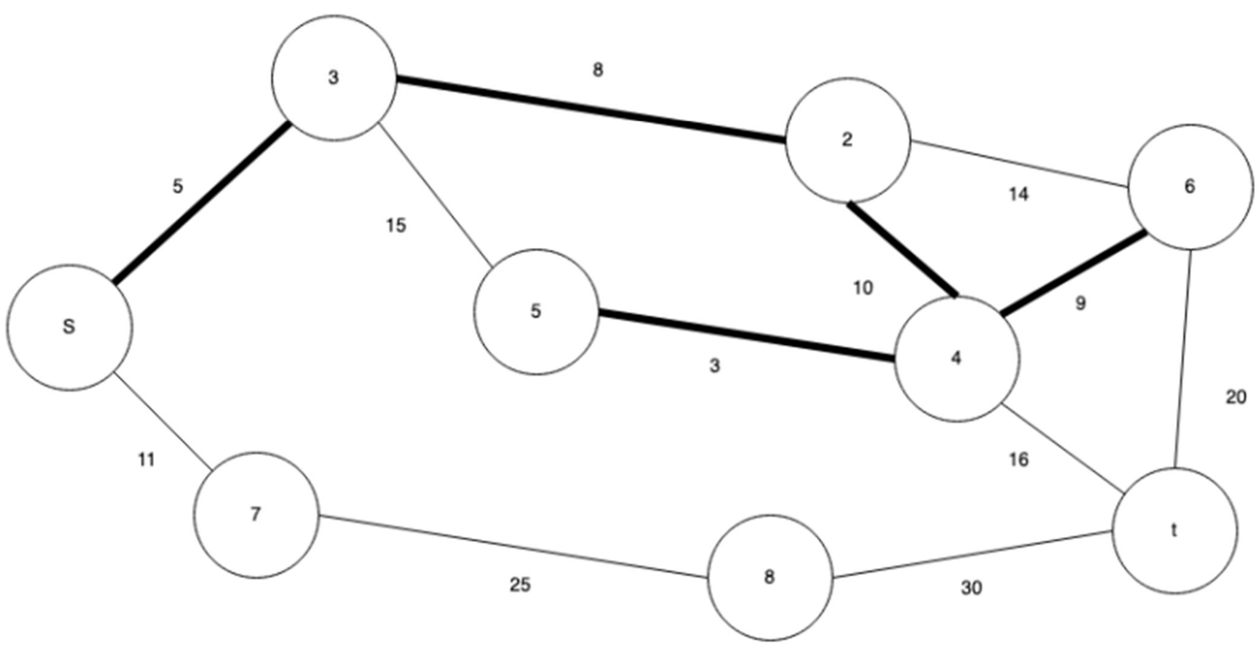
Step 4: -

From node ‘4’ selecting edge with weight 3



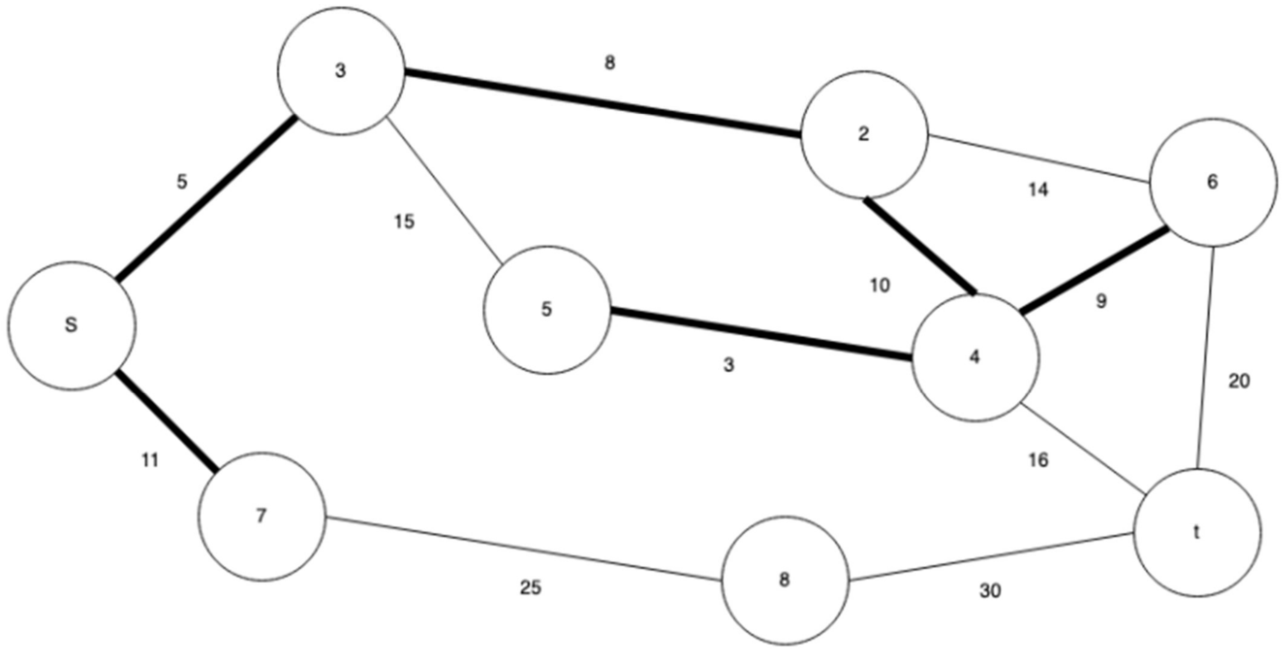

Step 5: -

From node ‘4’ selecting edge with weight 9



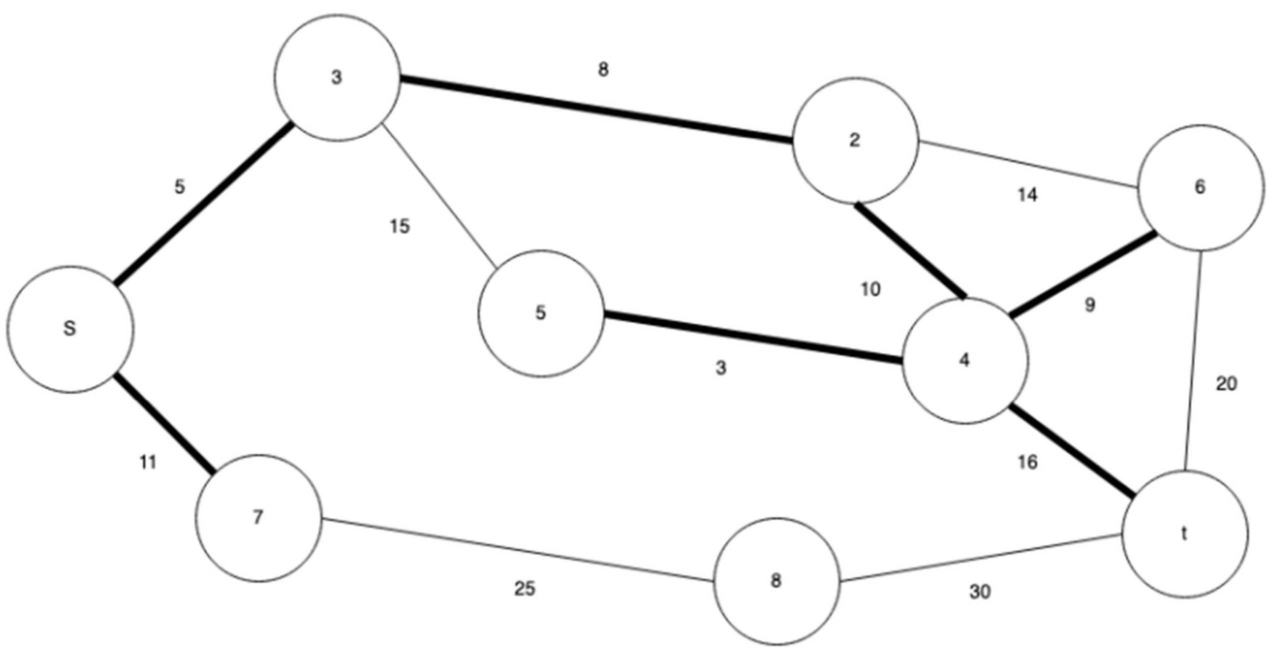
Step 6:

From node ‘s’ selecting the edge with weight 11



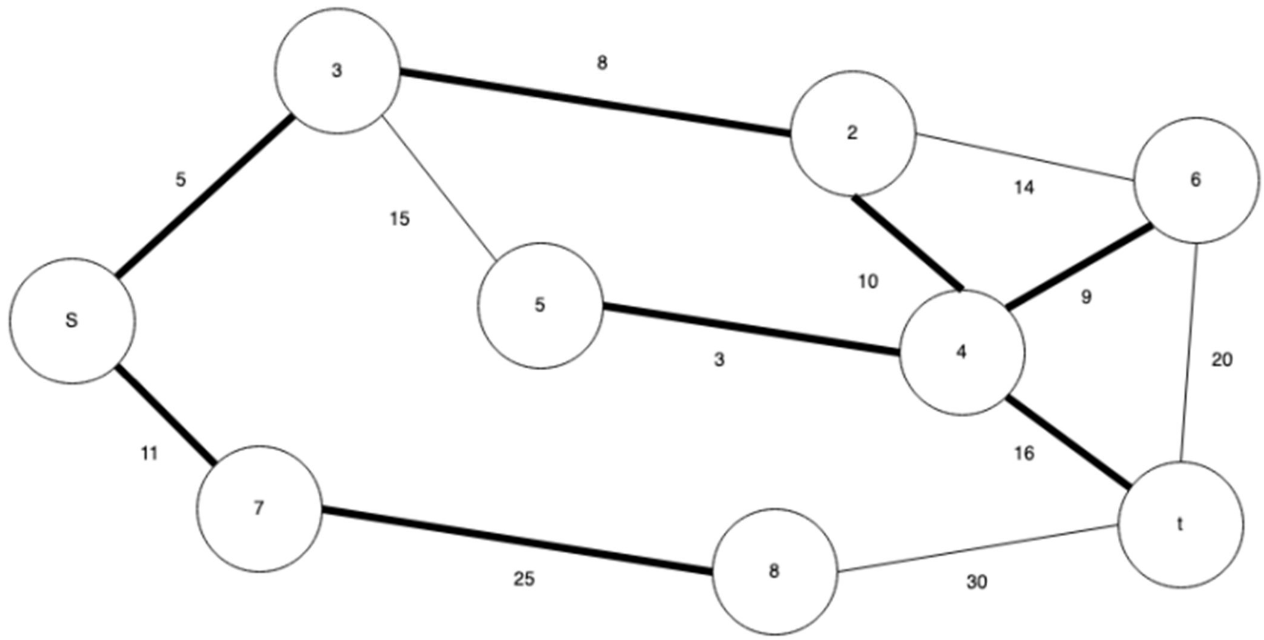

Step 7:

From node ‘4’ selecting edge with weight 16

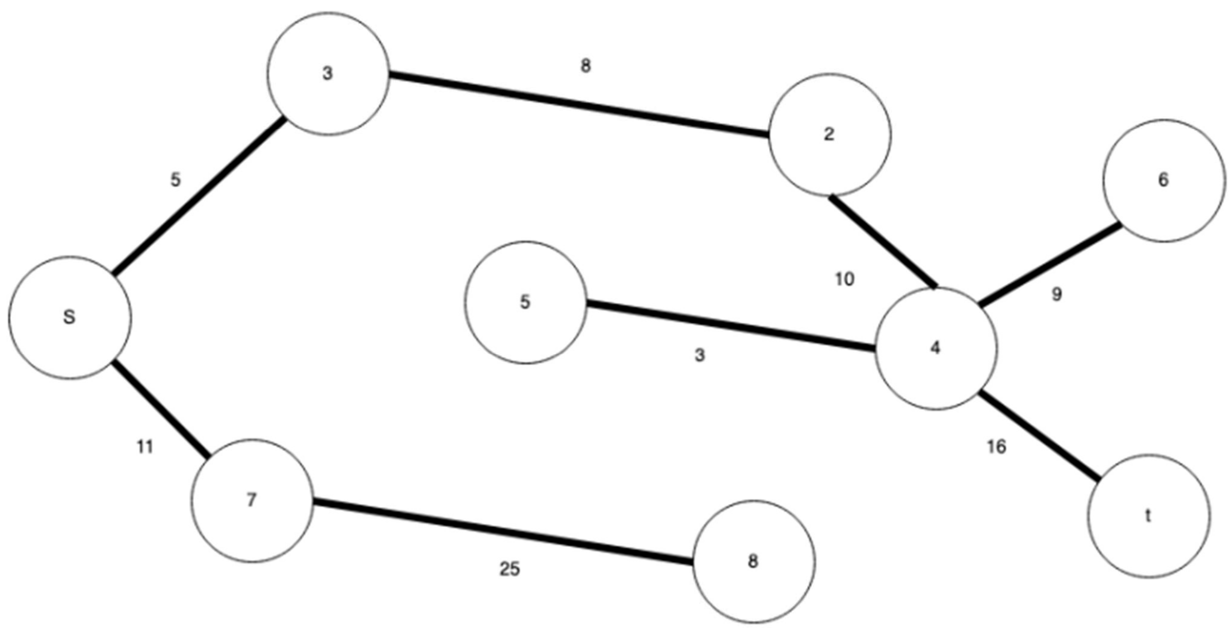


Step 8: -

From node '7' selecting the edge with weight 25




Final Minimum spanning tree




Total weight of minimum spanning tree = 5+8+10+3+9+11+16+25

= 87

'andya
Jenil

# Find MST
## Q.2 using Prim's, Kruskal, Reverse Delete

### 2 Finding Kuskal using MST Algorithm

| | Kruskal Order Edge rep. | Weight Edge |
|---|---|---|
| 1 | 4 → 5 | 3 |
| 2. | 5 → 3 | 5 |
| 3 | 3 → 2 | 8 |
| 4. | 4 → 6 | 9 |
| 5. | 2 → 4 | 10 |
| 6 | S → 7 | 11 |
| 7. | 4 → t | 16 |
| 8. | 7 → 8 | 25 |
| | weight total | $\boxed{87}$ |

$2 \xrightarrow{14} 6$ Forming cycle

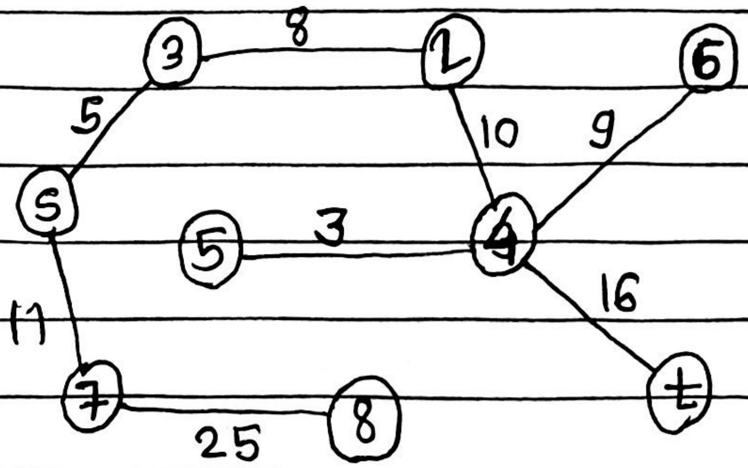
$3 \xrightarrow{15} 5$

So,

$6 \xrightarrow{20} t$ Discarded

$8 \xrightarrow{30} t$

Final Graph using Kruskal



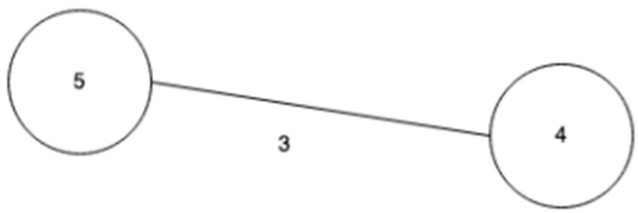

Kruskal's MST

weight total

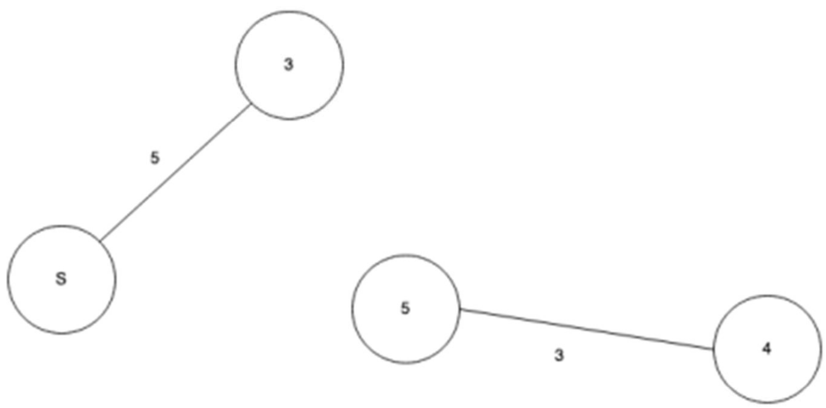
$\boxed{87}$

KRUSKAL's Algorithm

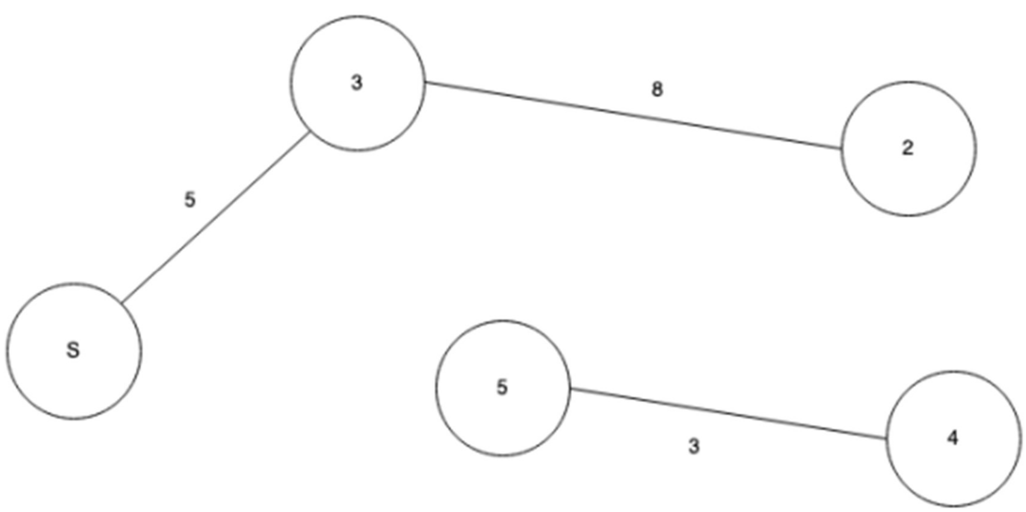
Selecting the Lowest edge with weight 3



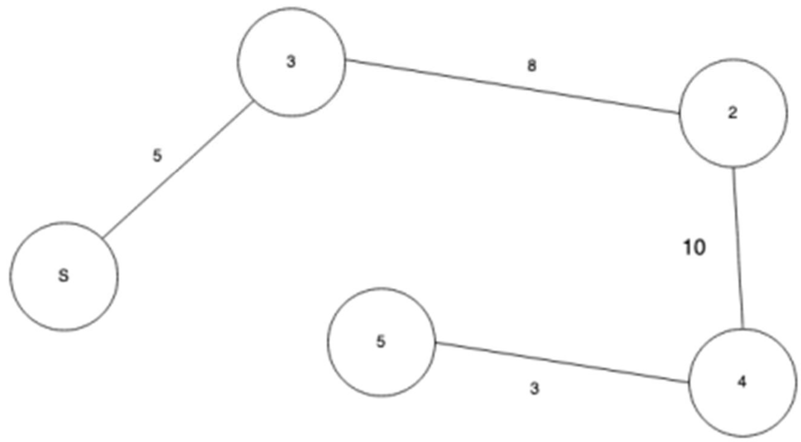

Step 2: -



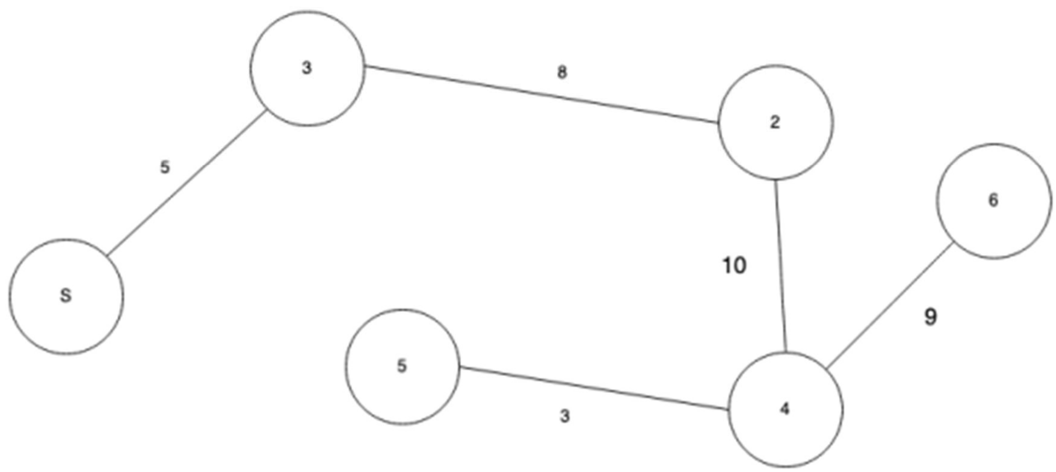

Step 3: -



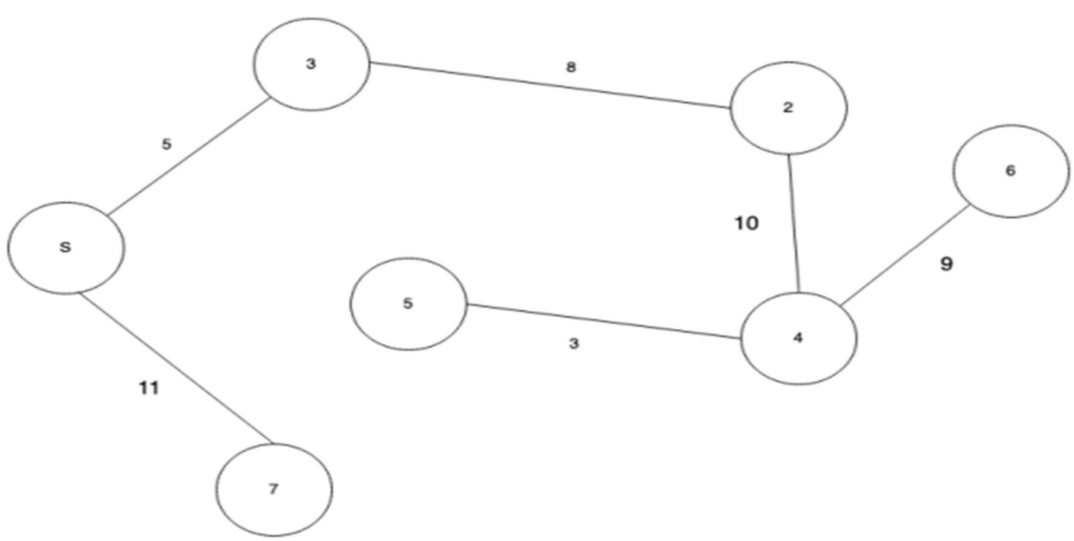
Step 4: -



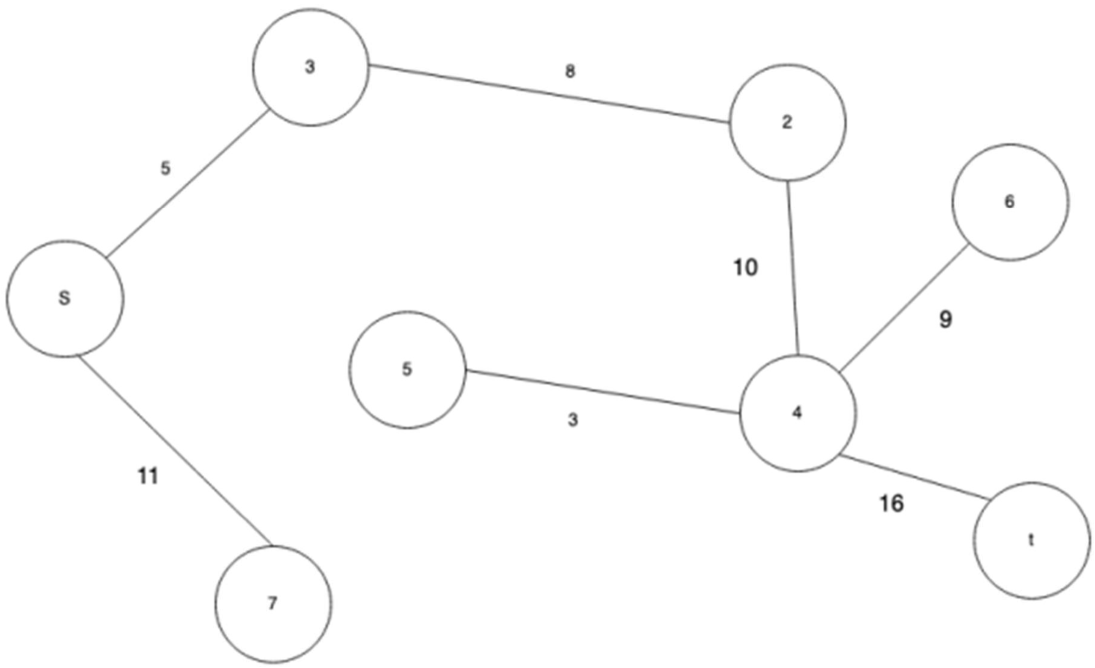

Step 5: -



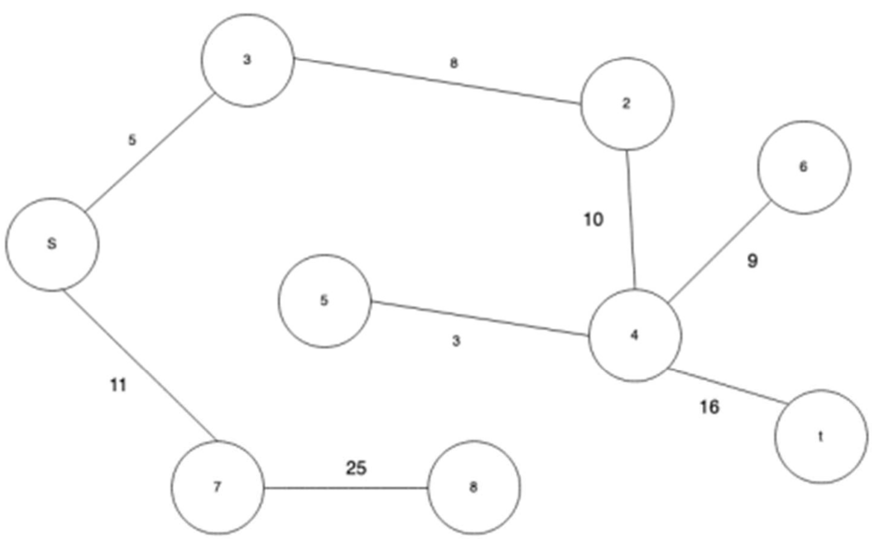

Step 6: -



Step 7: -




Step 8: -




Total minimum spanning tree weight = 3+5+8+9+10+11+16+25

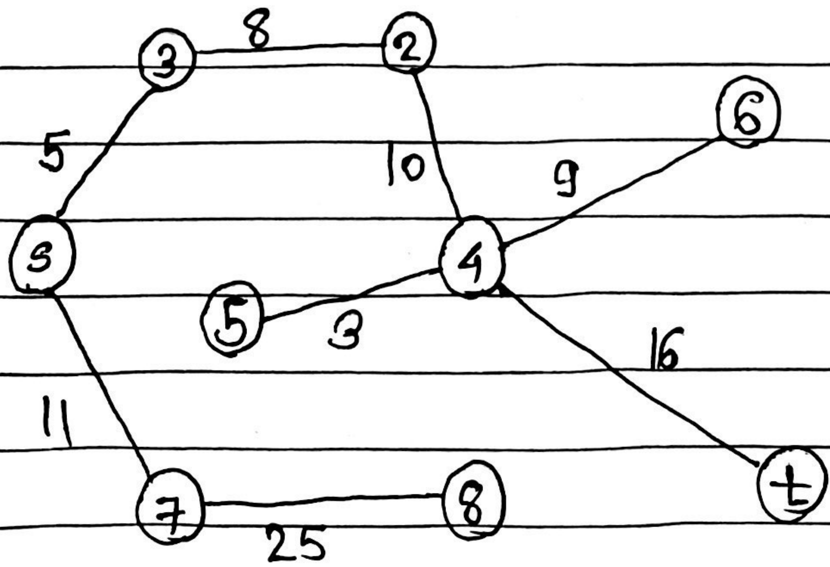
= 87

Jenil
Pandya

# Finding MST using Reverse Delete Algorithm

| Deleted Edges order | Edge weight |
|---|---|
| 8 → t | 30 |
| 6 → t | 20 |
| 2 → 6 | 14 |
| 3 → 5 | 15 |
| Removed weight | 79 |

## Final Graph After Removing Edges



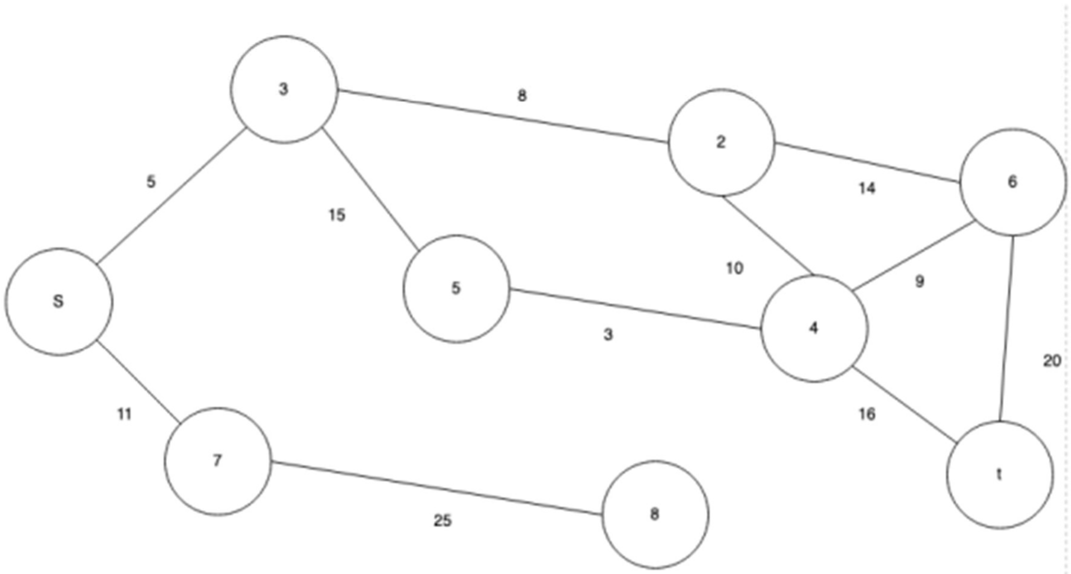

$5+8+10+9+3+16+11+25 = \boxed{87}$

MST weight 87

Reverse Delete Algorithm

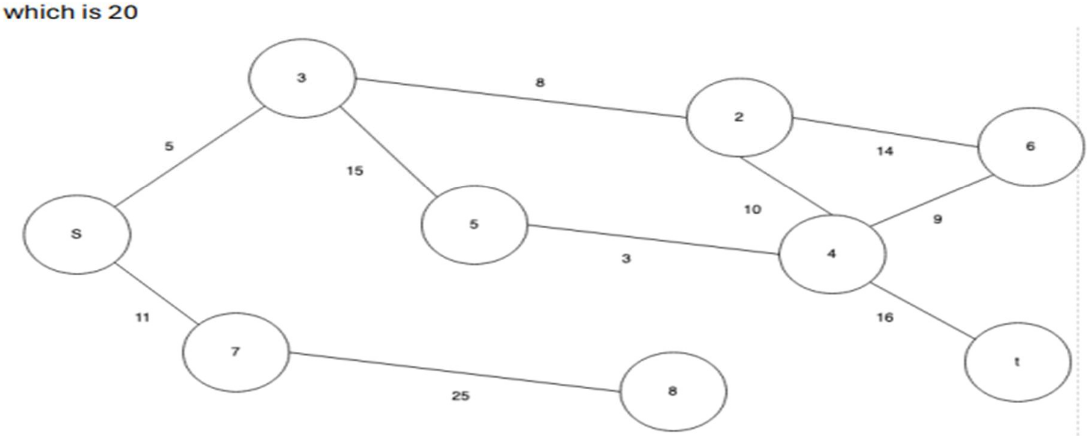
Step 1: -

Removing edge 30 as it's the highest.



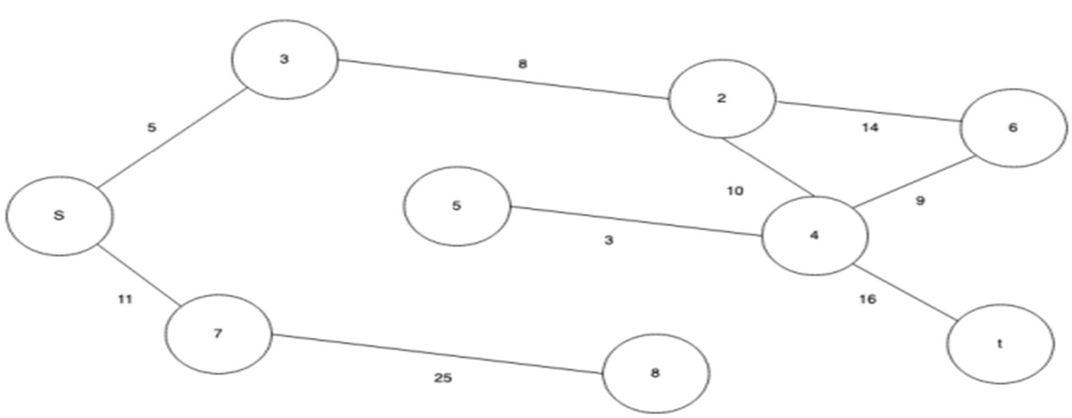

Step 2: -

The next highest weight is 25 but it can break the graph so we'll go for the next highest which is 20



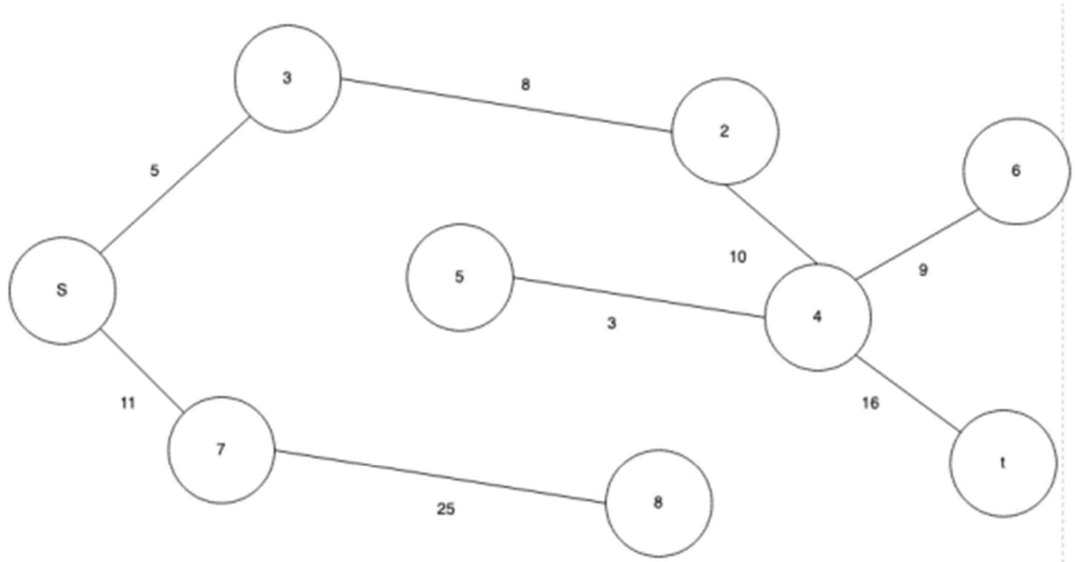

Step 3: -

The edge with weight 16 is the next highest but it can break the graph so selecting the next edge which is 15.



Step 4: -

Removing the highest edge 14




Weight of minimum spanning trees using Reverse Delete algorithm

= 5+8+10+3+9+11+16+25

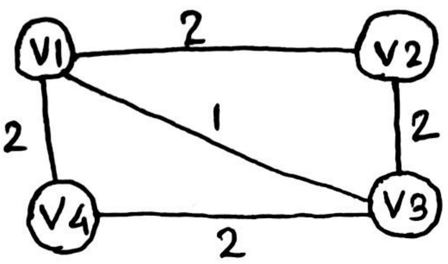
= 87

Jenil P.

**Q-3** Problem 22 of the Kleinberg and Tardos text

The four Nodes are V1, V2, V3, V4

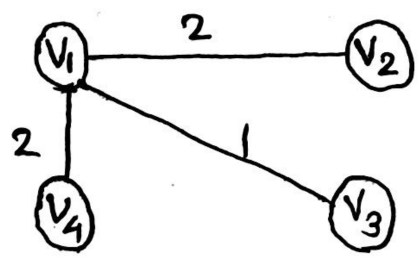
$(V_1, V_2)$ $(V_2, V_3)$ $(V_3, V_4)$ $(V_4, V_1)$ these edges have a cost of 2 and $(V_1, V_3)$ have cost of 1

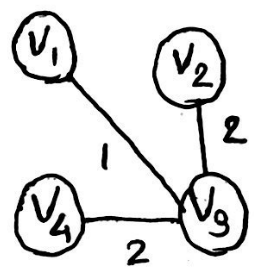


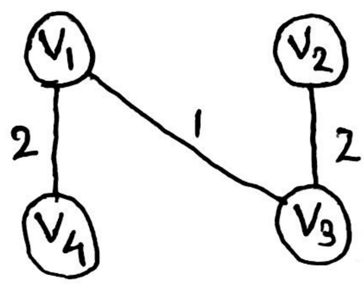

there exists many spanning tree for this graph where $c_e \geq 0$

the rule for a minimum spanning tree is that Number of edges = Numerbe of Vertices

In this graph we have 4, Vertices, these are possible MSTs



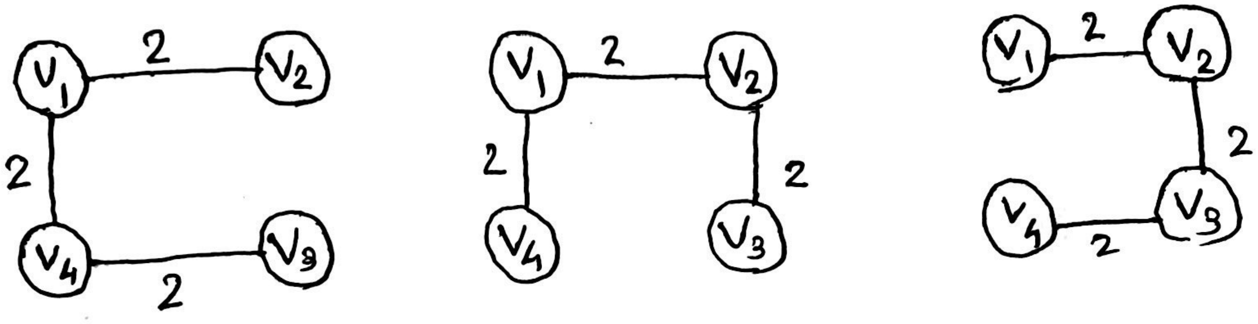




the total cost for All of these graph is 5

$2+2+1 = 5$

Consider spanning tree such that $e \in T$

Some minimum cost of graph




the minimum spanning tree weight for all the graph is 6, so we can say that 'T' is not a minimum spanning tree in G

Jenil P

Q.4

| $a_i$ | | | | |
|---|---|---|---|---|
| 4 | 6 | 9 | 14 | 17 |

| $a_j$ | | | | |
|---|---|---|---|---|
| 2 | 5 | 10 | 12 | 15 |

Step:1 $4 > 2$, $a_i > a_j$, 5 inversion

Aux. Array.

| | |
|---|---|
| 2 | |

Step:2

| $a_i$ | | | | |
|---|---|---|---|---|
| 4 | 6 | 9 | 14 | 17 |

| | $a_j$ | | | |
|---|---|---|---|---|
| 2 | 5 | 10 | 12 | 15 |

$4 < 5$, $a_i < a_j$, no inversion

Aux.

| | | |
|---|---|---|
| 2 | 4 | |

Step:3

| | $a_i$ | | | |
|---|---|---|---|---|
| 4 | 6 | 9 | 14 | 17 |

| | $a_j$ | | | |
|---|---|---|---|---|
| 2 | 5 | 10 | 12 | 15 |

$6 > 5$, $a_i > a_j$ 4 inversion

Aux Array

| | | | |
|---|---|---|---|
| 2 | 4 | 5 | |

Step:4

| | $a_i$ | | | |
|---|---|---|---|---|
| 4 | 6 | 9 | 14 | 17 |

| | | $a_j$ | | |
|---|---|---|---|---|
| 2 | 5 | 10 | 12 | 15 |

$6 < 10$, $a_i < a_j$ no inversion

Aux

| | | | | |
|---|---|---|---|---|
| 2 | 4 | 5 | 6 | |

Step: 5

| | | $a_i$ | | |
|---|---|---|---|---|
| 4 | 6 | 9 | 14 | 17 |

| | | $a_j$ | | |
|---|---|---|---|---|
| 2 | 5 | 10 | 12 | 15 |

$9 < 10$, $a_i < a_j$ no inversion

Aux

| | | | | | |
|---|---|---|---|---|---|
| 2 | 4 | 5 | 6 | 9 | |

Step: 6

| | | | $a_i$ | |
|---|---|---|---|---|
| 4 | 6 | 9 | 14 | 17 |

| | | $a_j$ | | |
|---|---|---|---|---|
| 2 | 5 | 10 | 12 | 15 |

$14 > 10$, $a_i > a_j$, 2 inversion

Aux

| | | | | | | |
|---|---|---|---|---|---|---|
| 2 | 4 | 5 | 6 | 9 | 10 | |

Step:-7

| | | | $a_i$ | |
|---|---|---|---|---|
| 4 | 6 | 9 | 14 | 17 |

| | | | $a_j$ | |
|---|---|---|---|---|
| 2 | 5 | 10 | 12 | 15 |

$14 > 12$, $a_i > a_j$, 2 inversion

Aux

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2 | 4 | 5 | 6 | 9 | 10 | 12 | |

**Step:8**

| | | | $a_i$ | |
|---|---|---|---|---|
| 4 | 6 | 9 | 14 | 17 |

| | | | | $a_j$ |
|---|---|---|---|---|
| 2 | 5 | 10 | 12 | 15 |

$14 < 15$, $a_i < a_j$, no inversions, aux

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2 | 4 | 5 | 6 | 9 | 10 | 12 | 14 |

**Step:9**

| | | | | $a_i$ |
|---|---|---|---|---|
| 4 | 6 | 9 | 14 | 17 |

| | | | | $a_j$ |
|---|---|---|---|---|
| 2 | 5 | 10 | 12 | 15 |

$17 > 15$, $a_i > a_j$, 1 [redacted] inversions,

Aux. Array

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 4 | 5 | 6 | 9 | 10 | 12 | 14 | 15 | |

**Step: 10** No Inversion Add 17

Final Merged

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 4 | 5 | 6 | 9 | 10 | 12 | 14 | 15 | 17 |

Count Inversions :- $5 + 4 + 2 + 2 + 1 = 14$

total 14 inversions

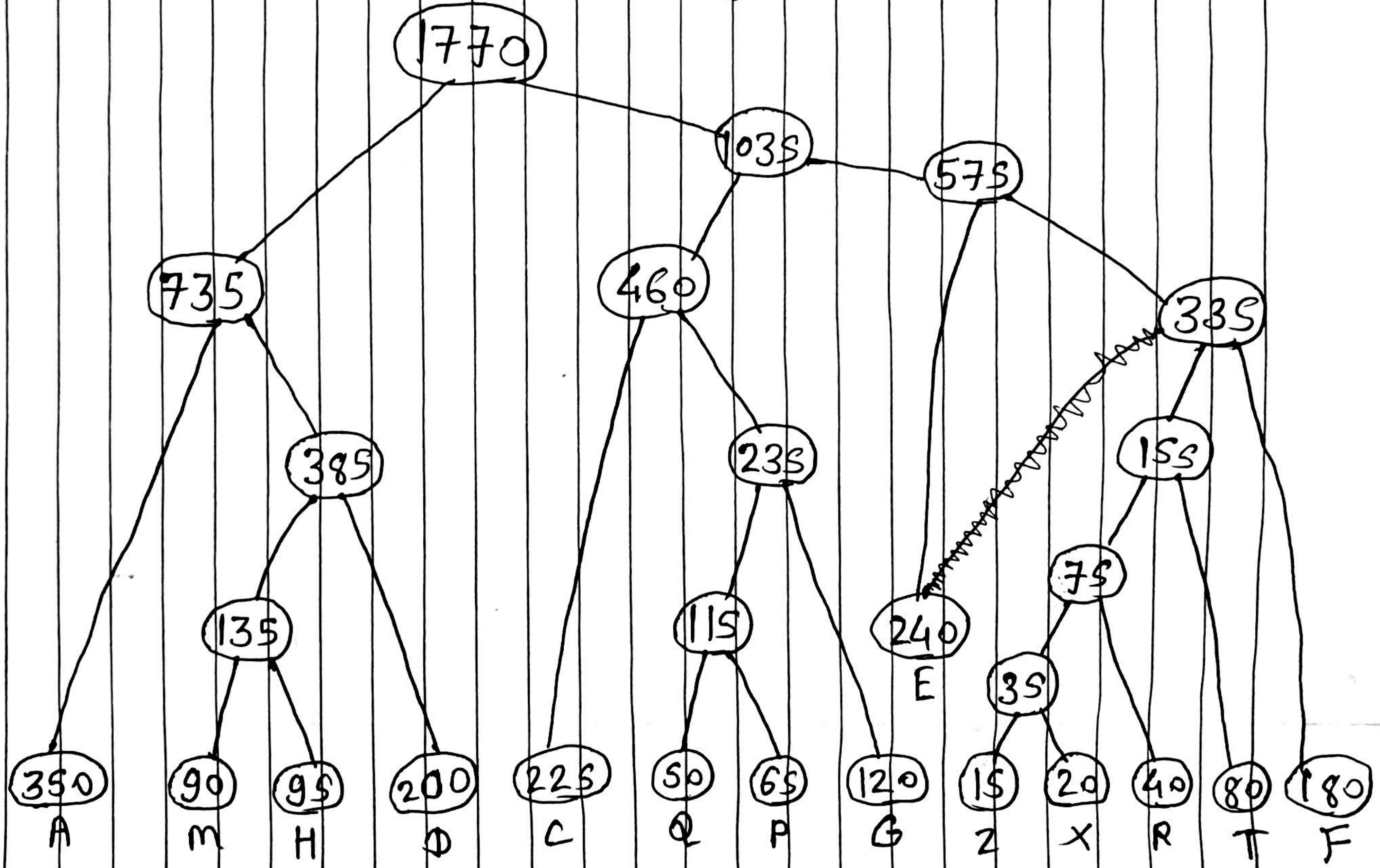
5.

| Charachter | Frequency |
|---|---|
| A | 350 |
| E | 240 |
| C | 225 |
| D | 200 |
| F | 180 |
| G | 120 |
| H | 95 |
| M | 90 |
| T | 80 |
| P | 65 |
| Q | 50 |
| R | 40 |
| X | 20 |
| Z | 15 |

Step1: Create Priority queue of frequency

| Z | X | R | Q | P | T | M | H | G | F | D | C | E | A |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 20 | 40 | 50 | 65 | 80 | 90 | 95 | 120 | 180 | 200 | 225 | 240 | 350 |

Jenil Pandya



| Charachter | freq | Huffman code | 4-bits fixed-code |
| --- | --- | --- | --- |
| A | 350 | 00 | 0000 |
| E | 240 | 110 | 0001 |
| C | 225 | 100 | 0010 |
| D | 200 | 011 | 0011 |
| F | 180 | 1111 | 0100 |
| G | 120 | 1011 | 0101 |
| H | 95 | 0101 | 0110 |
| M | 90 | 0100 | 0111 |
| T | 80 | 11101 | 1000 |
| P | 65 | 10101 | 1001 |
| Q | 50 | 10100 | 1010 |
| R | 40 | 111001 | 1011 |
| X | 20 | 1110001 | 1100 |
| Z | 15 | 1110000 | 1101 |

$$ABL = \frac{\Sigma(f \times \text{code length})}{\text{total freq}}$$

4 bits Fixed code

$$ABL = 4(350+240+225+200+ \dots +20+15)/1770$$

$$= 4 \cdot \frac{1770}{1770}$$

$\boxed{ABL = 4 \text{ bits/Letter}}$

Huffman

$$ABL = \frac{[350\times2 + 250\times3 + 225\times3 + \dots + 20\times7 + 15\times7]}{1770}$$

$$= \frac{6095}{1770}$$

$\boxed{ABL = 3.44 \text{ bits/Letter}}$