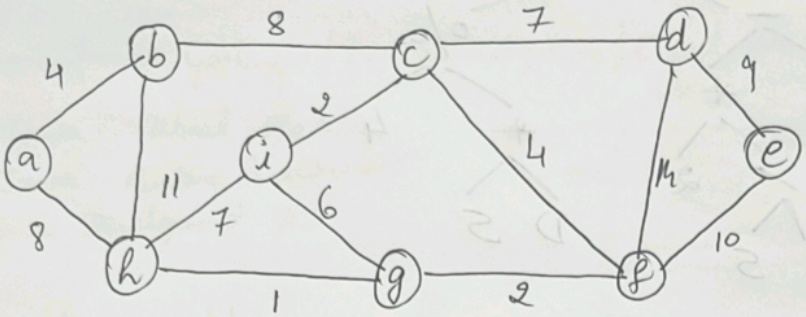
In the given graph, visit the nodes with their weights using Kruskal's algorithm. Traverse it step by step.

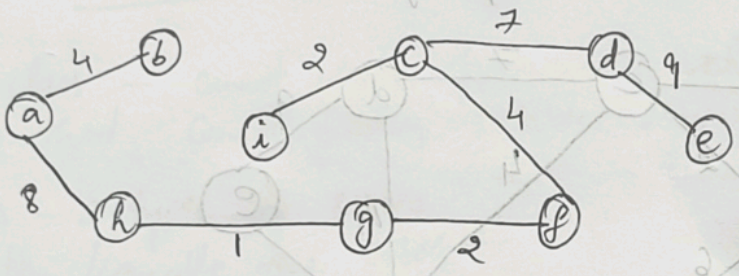



Step1

| ab | ah | bc | bh | cd | ci | cf | de | df | ef | gf | gh | gi | hi |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 8 | 8 | 11 | 7 | 2 | 4 | 9 | 14 | 10 | 2 | 1 | 6 | 7 |

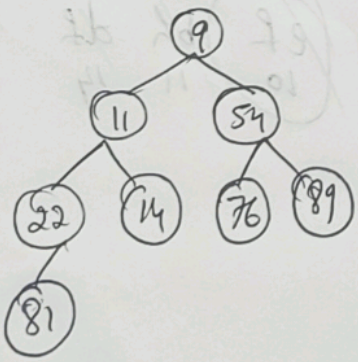
Step2

| gh | ci | fg | ab | ef | gi | hi | cd | bc | ah | de | ef | bh | df |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



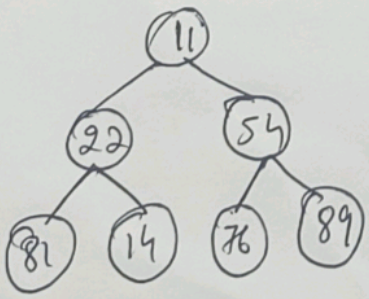

## Minimum heapSort

Initial array [81, 89, 9, 11, 14, 76, 54, 22]



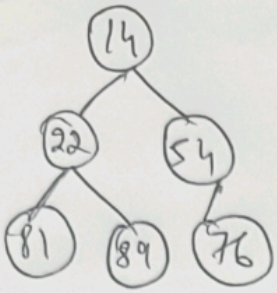

[81, 11, 54, 22, 14, 76, 89, 9]

Heapify - [81, 11, 54, 22, 14, 76, 89]



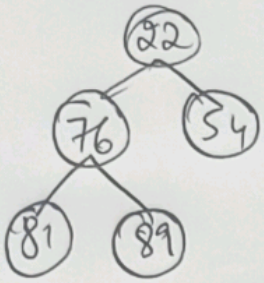

[89, 22, 54, 81, 14, 76, 11, 9]

Heapify - [89, 22, 54, 81, 14, 76]



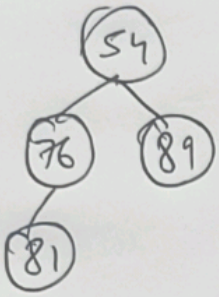

[76, 22, 54, 81, 89, 14, 11, 9]

Heapify - [76, 22, 54, 81, 89]




[89, 76, 54, 81, 22, 14, 11, 9]

Heapify - [89, 76, 54, 81]




[81, 76, 89, 54, 22, 14, 11, 9]

Heapify - [81, 76, 89]

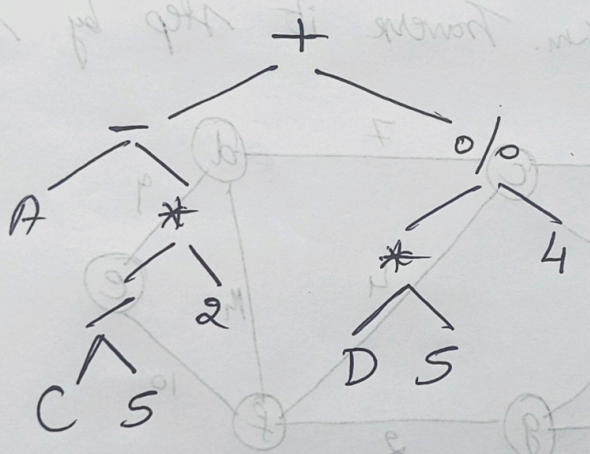

[89, 81, 76, 54, 22, 14, 11, 9]

Heapify - [89, 81]


[89, 81, 76, 54, 22, 14, 11, 9]

Sorted array - [9, 11, 14, 22, 54, 76, 81, 89]

1) Traverse the given expression tree with inorder, preorder & postorder.




Inorder Traversal : A - C / 5 * 2 + D * 5 % 4

Preorder Traversal : + - A * / C 5 2 % * D 5 4

Postorder Traversal : A C 5 / 2 * - D 5 * 4 % +