

Huffman Coding

zip

↳ Data compression

1 Byte — 8 bits

Pre-requisite

↳ Heap

Data structure

Input →

character ← freq

→

a = 5

b = 9

c = 12

d = 13

e = 16

f = 45

$$5 \times 8 + 9 \times 8 + 12 \times 8 + 13 \times 8 + 16 \times 8 + 45 \times 8$$

$$\Rightarrow 40 + 72 + 96 + 104 + 128 + 360$$

\Rightarrow 800 bits

Pseudocode → $\left\{ \begin{array}{l} \text{More frequency} \rightarrow \text{smaller bit} \\ \text{Low frequency} \rightarrow \text{larger bit} \end{array} \right.$
Huffman Tree

1) Build heap $\text{————— } O(n)$

↳ Minheap

↳ Mathematical

2) Pop 2 times & insert the addition of two deleted elements \rightarrow Minheap

Derivation
dition $\frac{2 \text{ Del}}{1 \text{ Insertion}}$
(n-1) 3 log n

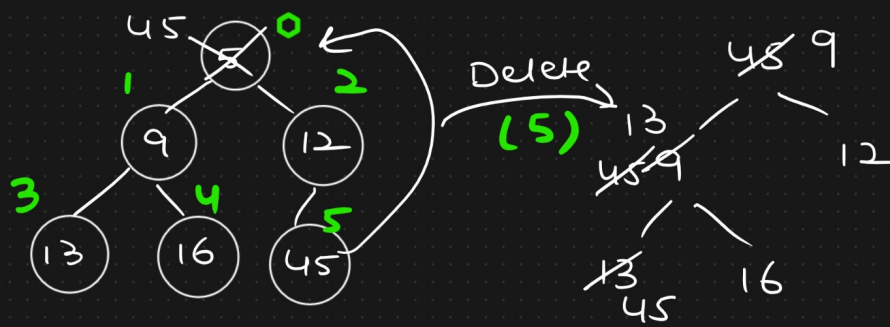
Deletion $\text{---} O(\log n)$

→ Minheap

↳ smallest

element-

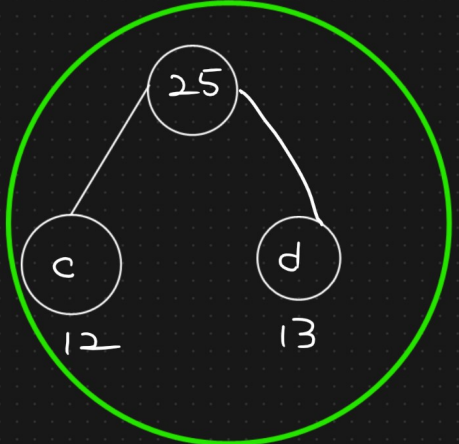
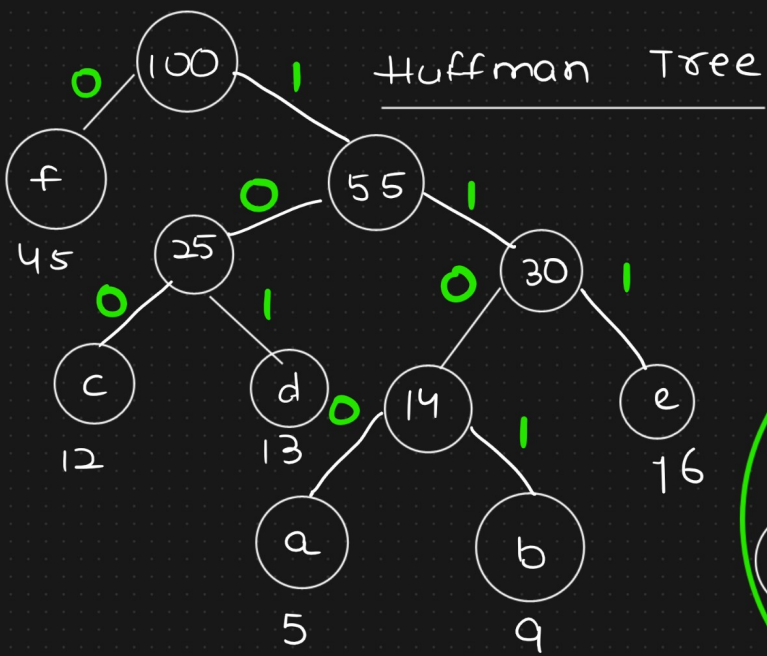
Insertion $\underline{\hspace{2cm}}$ $O(\log n)$



$(\checkmark 45, 16, \checkmark 14)$
 $\underline{25, 45, 16, 14}$

Output

Delete
 f \rightarrow 0
 a \rightarrow 1100
 b \rightarrow 1101
 c \rightarrow 100
 d \rightarrow 101
 e \rightarrow 111



Time complexity

$$\Rightarrow n + (n-1) * 3 \log n$$

$$\Rightarrow \underline{\underline{O(n \log n)}}$$

Python



heappush
(Insertion)

heappop (Deletion)