🔷 Example Walkthrough

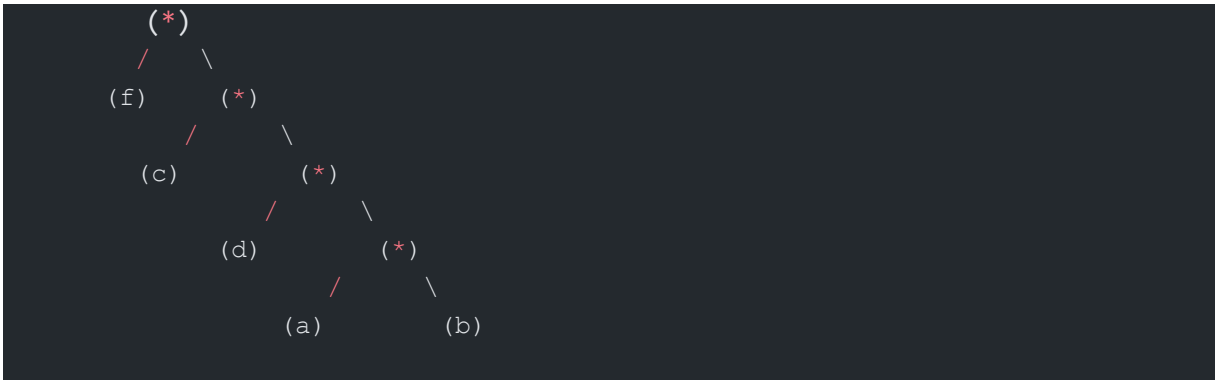
🔷 Step 1: Initial Min-Heap

```
[(5, 'a'), (9, 'b'), (12, 'c'), (13, 'd'), (16, 'e'), (45, 'f')]
```

🔷 Step 2: Merging Nodes (Building Huffman Tree)

Merge Step	Left (0)	Right (1)	New Node
1	a (5)	b (9)	(14, "ab")
2	c (12)	d (13)	(25, "cd")
3	ab (14)	e (16)	(30, "abe")
4	cd (25)	abe (30)	(55, "cdabe")
5	f (45)	cdabe (55)	(100, "fcdaeb") **(Root Node)**

🔷 Step 3: Final Huffman Tree



🔷 Step 4: Printing Huffman Codes

📌 Observations:

- **Most frequent character (f)** gets the **shortest code (0).**
- **Least frequent character (a)** gets a **longer code (1110).**

Merging Nodes (Building the Tree)

Step 1: Merge 'a' and 'b'

```
New Node: ('ab', 14) → Left: 'a' (5), Right: 'b' (9)
```

Heap now contains:

```
[(12, 'c'), (13, 'd'), (14, 'ab'), (16, 'e'), (45, 'f')]
```

Step 5: Merge "f" (45) and "cdabe" (55) → ROOT NODE

```
New Node: ('fcdaeb', 100) → Left: 'f' (45), Right: "cdabe" (55)
```

🔷 Final Huffman Tree