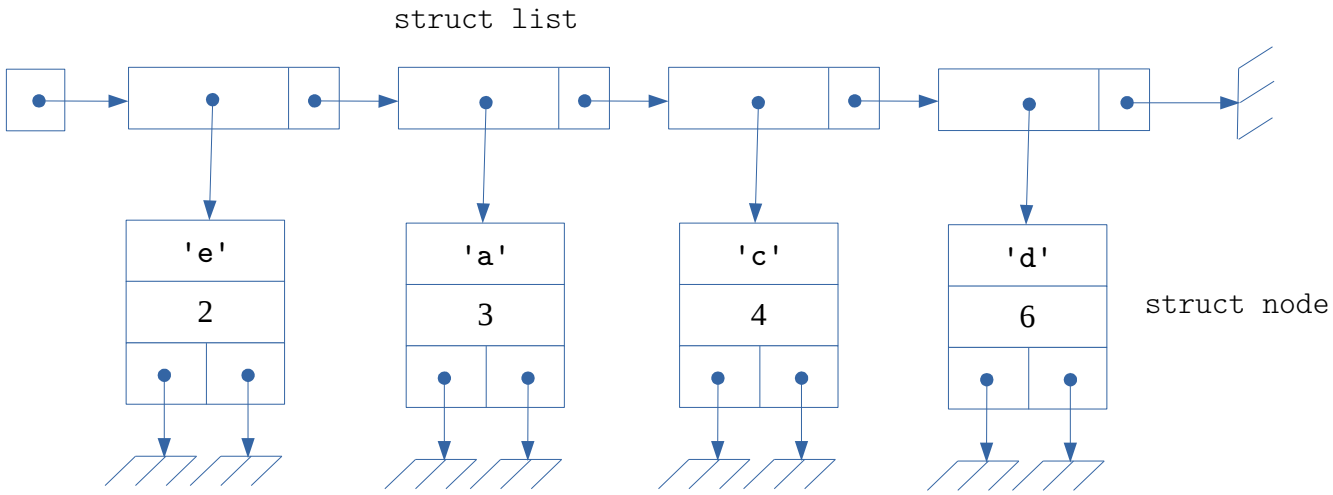
Si on saisit le texte :
aaaccccddddddee

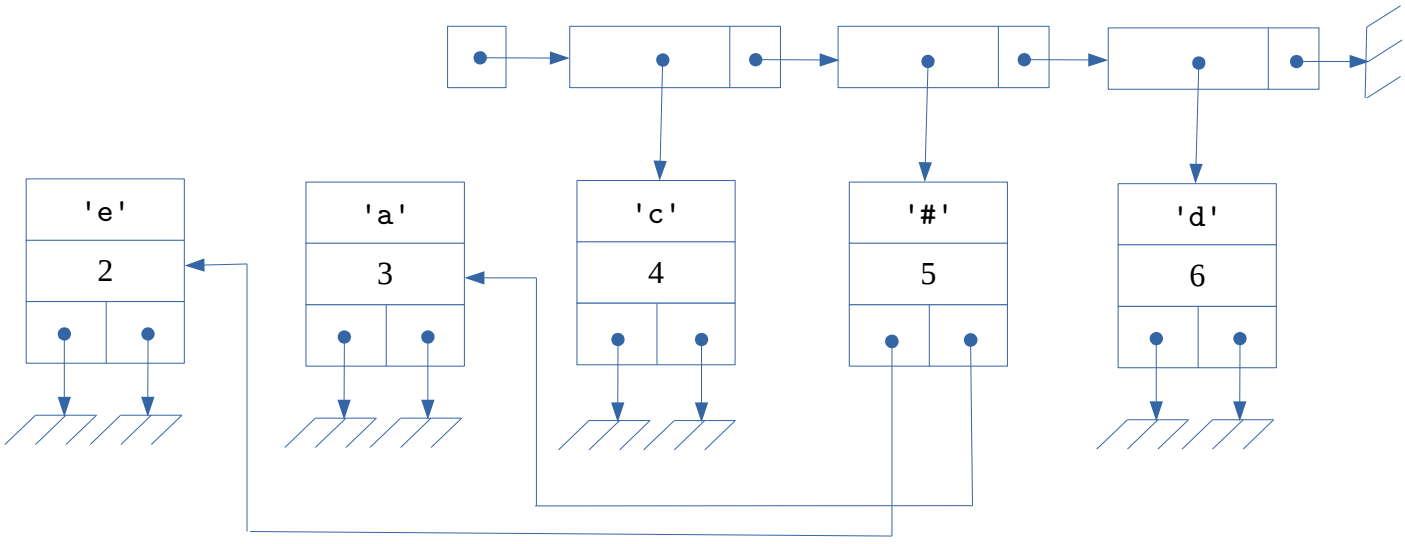
```
struct node {
  char c; // '#' for internal nodes
  int count;
  struct node *left;
  struct node *right;
};

struct list {
  struct node *data;
  struct list *next;
};
```

Liste avant contruction de l'arbre :



Contruction de l'arbre en cours (après la première itération) :

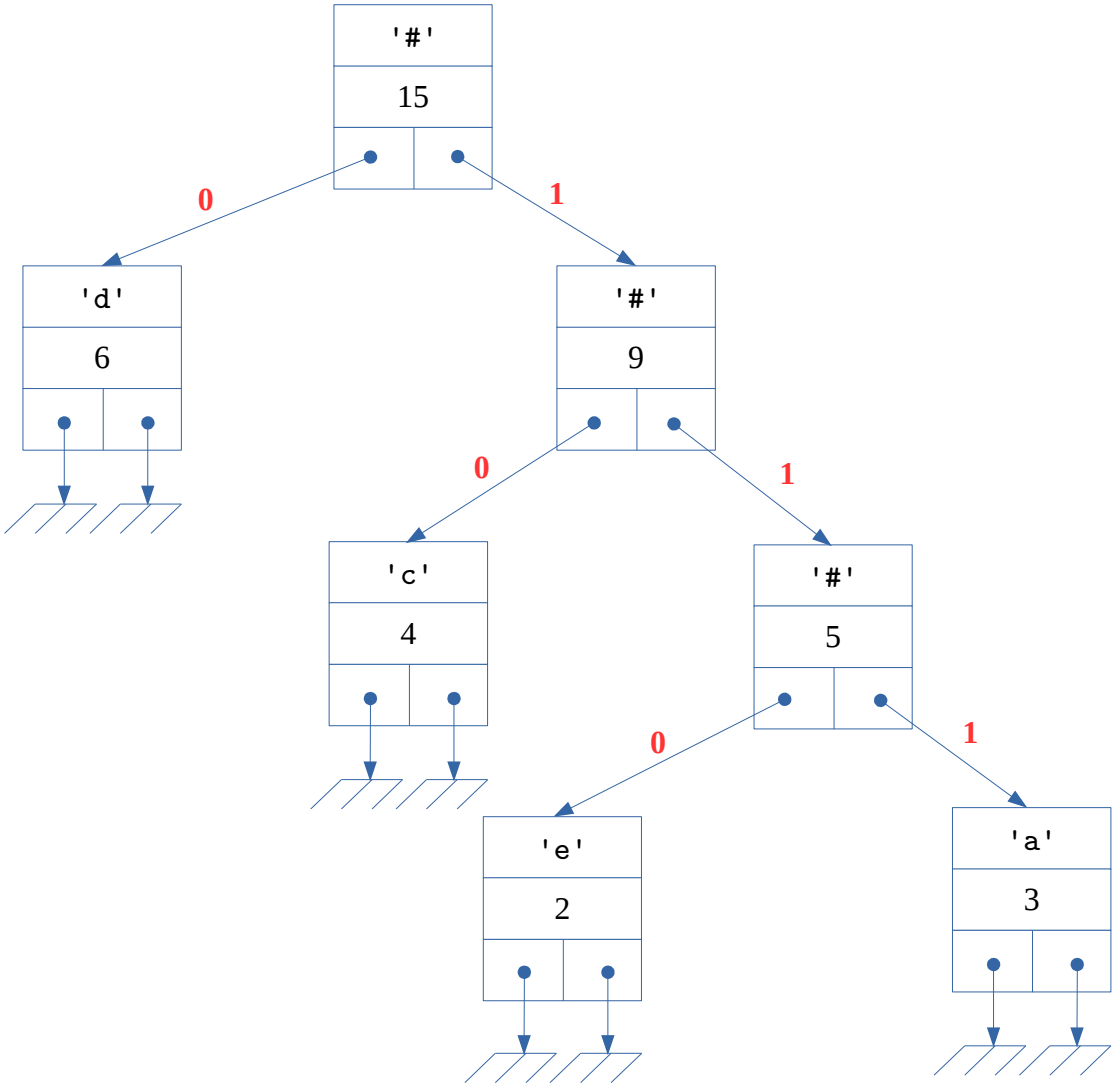


Etapes de construction de l'arbre :

Etape 0 :	e2 a3 c4 d6	donne c4 #5 d6
Etape 1 :	c4 #5 d6	donne d6 #9
Etape 2 :	d6 #9	donne #15

A chaque étape, le noeud en rouge, qui est ajouté, est le père des noeuds en bleu, qui sont supprimés. La liste résultante est toujours triée.
La liste finale contient un seul noeud dont le champ `data` pointe sur la racine de l'arbre de Huffman.

Arbre terminé :



On en déduit le code des caractères (présents dans le texte à compresser) :
d : 0
c : 10
e : 110
a : 111

Rappel du texte initial :
`aaaccccddddddee`
Résultat de la compression :
`11111111110101010000000110110`