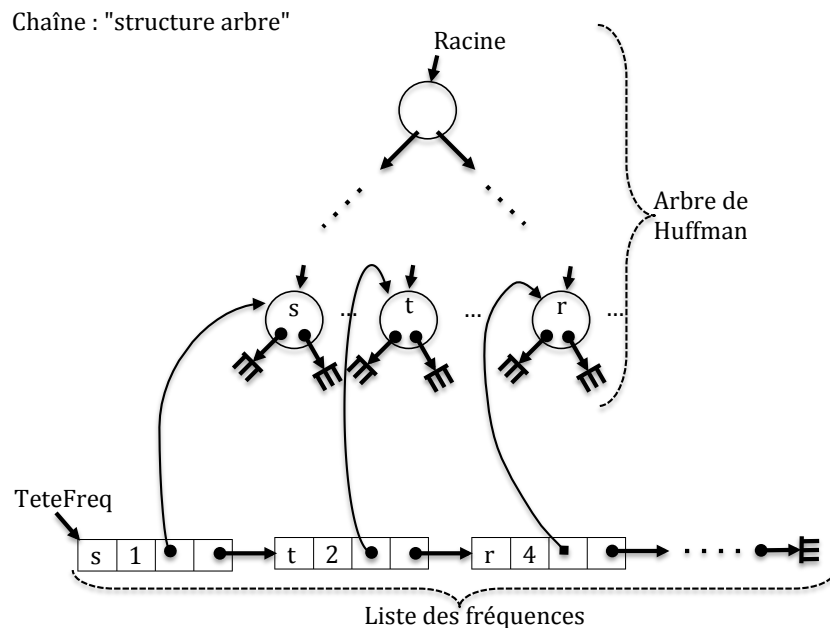


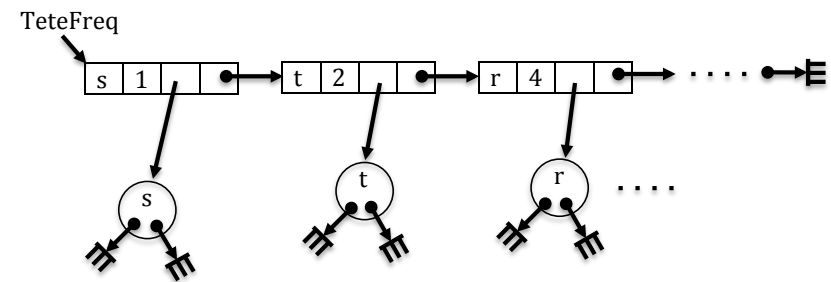
Examen de rattrapage

On souhaite écrire les algorithmes permettant de créer l'arbre de Huffman correspondant à une chaîne de caractères donnée, tel que vu dans le cours. Pour cela, on vous propose de commencer par créer une liste linéaire chaînée contenant les caractères de la chaîne à coder avec leurs fréquences d'apparition et des pointeurs vers leurs noeuds correspondants dans l'arbre de Huffman, tel que représenté dans la figure suivante :



Initialement, l'arbre et la liste des fréquences sont vides. Pour réaliser ce travail, on vous demande de :

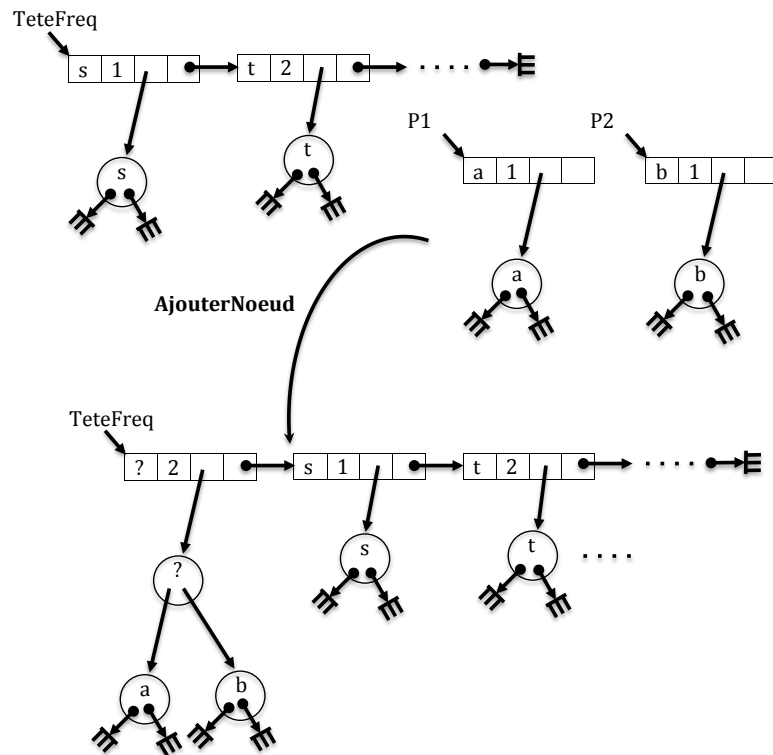
1. Donner les structures de données nécessaires à la représentation en mémoire de la liste des fréquences et de l'arbre de Huffman avec les initialisations nécessaires.
2. Ecrire la procédure **AjouterCar(C :caractère)** permettant d'ajouter le caractère **C** à la liste des fréquences, avec un pointeur nil au noeud de l'arbre. (si **C** existe déjà, on incrémente uniquement sa fréquence)
3. Ecrire la procédure **CreerListeFreq(Ch :chaîne)** permettant de créer la liste de tête TeteFreq, contenant les caractères de la chaîne **Ch** avec leurs fréquences et des pointeurs nil vers l'arbre, en utilisant la procédure précédente.
4. Ecrire la procédure **CreerFeuilles** permettant de créer, pour chaque caractère de la liste des fréquences, un noeud feuille de l'arbre contenant ce caractère, tel que présenté dans la figure suivante :



5. Ecrire la fonction **GetMinFreq** permettant de retourner le pointeur vers le caractère de fréquence minimale dans la liste des fréquences tout en le supprimant de la liste **sans le libérer**.
6. Ecrire la procédure **AjouterNoeud(P1,P2 :Pointeur(TMaignon))** permettant de :

- (a) Créer un nouveau nœud de l'arbre ayant deux fils, les nœuds pointés par P1 et P2,
- (b) Ajouter un nouveau maillon à la liste des fréquences avec une fréquence égale à la somme des fréquences de P1 et P2, et avec un pointeur vers le nouveau nœud créé de l'arbre.

Exemple :



Barème : 2 + 3 + 3 + 3 + 3 + 3 + 3

★★★ Bonne chance ★★★

A.Djeffal

7. Ecrire la procédure **CreerArbreHuffman**(Ch :chaîne) permettant de créer l'arbre de Huffman correspondant à la chaîne de caractères **Ch** en utilisant les procédures et la fonction précédentes.

Corrigé type

Exercice 1 :

1. Structures de données (2 pts)

```
Type TNoeud = Structure
  Car : caractère;
  FG,FG : Pointeur(TNoeud);
Fin ;

Type TMaillon = Structure
  Car : caractère;
  Freq : entier;
  Noeud : Pointeur(TNoeud);
  Suivant : Pointeur(TMaillon);
Fin ;

Var Racine : Pointeur(TNoeud); Initialisée Nil
  TeteFreq : Pointeur(TMaillon); Initialisée Nil
```

2. Procédure AjouterCar (3 pts)

```
Procédure AjouterCar( C : caractère);
Var P : Pointeur(TMaillon);
Début
  P ← TeteFreq;
  Tant que ((P ≠ Nil) et (Car(P) ≠ C)) faire
    | P ← Suivant(P);
  Fin TQ;
  Si (P ≠ Nil) Alors
    | Aff.Freq(P, Freq(P)+1)
  Sinon
    | Allouer(P);
    | Aff_Car(P, C);
    | Aff_Freq(P, 1);
    | Aff_Noeud(P, Nil);
    | Aff_Adr(P, TeteFreq);
    | TeteFreq ← P;
  Fin Si;
Fin;
```

3. Procédure CreerListeFreq (3 pts)

```
Procédure CreerListeFreq( Ch : chaîne);  
Var P : Pointeur(TMaignon);  
Début  
  TeteFreq ← Nil;  
  Pour i de 1 à Longueur(Ch) faire  
    AjouterCar(Ch[i])  
  Fin Pour;  
Fin;
```

4. Procédure CeerFeuilles (3 pts)

```
Procédure CeerFeuilles();  
Var P : Pointeur(TMaignon);  
      N : Pointeur(TNoeud);  
Début  
  P ← TeteFreq;  
  Tant que ((P ≠ Nil)) faire  
    Allouer(N);  
    Aff_Car(N, Car(P));  
    Aff_FG(N, Nil);  
    Aff_FD(N, Nil);  
    Aff_Noeud(P, N);  
  Fin TQ;  
Fin;
```

5. Fonction GetMinFreq (3 pts)

```
Fonction GetMinFreq() : Pointeur(TMaignon);  
Var P, Prec, PrecMin, PMin : Pointeur(TMaignon);  
Début  
  P ← TeteFreq;  
  PMin ← P;  
  PrecMin ← Nil;  
  Prec ← Nil;  
  Tant que ((P ≠ Nil)) faire  
    Si (Freq(P) < Freq(PMin)) Alors  
      PMin ← P;  
      PrecMin ← Prec;  
    Fin Si;  
    Prec ← P;  
    P ← Suivant(P);  
  Fin TQ;  
  Si (PrecMin ≠ Nil) Alors  
    Aff_Adr(PrecMin, Suivant(P));  
  Sinon  
    Si (TeteFreq ≠ Nil) Alors  
      TeteFreq ← Suivant(TeteFreq);  
    Fin Si;  
  Fin Si;  
  GetMinFreq ← PMin;  
Fin;
```

6. Procédure AjouterNoeud (3 pts)

```
Procédure AjouterNoeud( P1, P2 : Pointeur(TMaillo));  
Var P : Pointeur(TMaillo);  
      N : Pointeur(TNoeud);  
Début  
  Allouer(N);  
  Aff_Car(N, '?');  
  Aff_FG(N, Noeud(P1));  
  Aff_FD(N, Noeud(P2));  
  Allouer(P);  
  Aff_Car(P, '?');  
  Aff_Freq(P, Freq(P1) + Freq(P2));  
  Aff_Noeud(P, N);  
  Aff_Adr(P, TeteFreq);  
  TeteFreq ← P;  
Fin;
```

7. Procédure CreerArbreHuffman (3 pts)

```
Procédure CreerArbreHuffman( Ch : chaîne);  
Var P1, P2 : Pointeur(TMaillo);  
Début  
  CreerListeFreq(Ch);  
  CreerFeuilles;  
  Tant que ((TeteFreq ≠ Nil) et (Suivant(TeteFreq) ≠  
  Nil)) faire  
    P1 ← GetMinFreq;  
    P2 ← GetMinFreq;  
    AjouterNoeud(P1, P2);  
    Libérer(P1);  
    Libérer(P2);  
  Fin TQ;  
  Si (TeteFreq ≠ Nil) Alors  
    Racine ← Noeud(TeteFreq);  
    Libérer(TeteFreq);  
  Sinon  
    Racine ← Nil;  
  Fin Si;  
Fin;
```