

ESP de Nouakchott

TD et TP d'algorithmique

20 novembre 2014

1 Énoncés

1. (Devoir 2013-2014) L'objectif de cette exercice est de définir un modèle pour une structure chaînée particulière. On veut implémenter une liste de manière à ce que la tête de la liste ainsi que l'élément du milieu soient accessibles directement via des pointeurs. Pour faciliter l'usage de cette liste on doit aussi garder le nombre d'éléments de la liste dans une variable que l'on appelle N (voir Figure 4)
De plus, cette liste doit être toujours triée dans l'ordre croissant de ces éléments (pour simplifier on suppose que les éléments sont de type Entier). Les opérations doivent donc respecter cette contrainte.
 - (a) Écrire le modèle de données correspondant à cette structure
 - (b) Écrire un algorithme correspondant à chacune des opérations suivantes (essayer d'exploiter la structure de la liste pour optimiser vos algorithmes) :
 - i. Create : $\text{Entier} \longrightarrow \text{Liste}$
Cette opération crée une liste qui contient un seul élément envoyé comme paramètre d'entrée
 - ii. Add : $\text{Liste} \times \text{Entier} \longrightarrow \text{Liste}$
Cette opération ajoute un élément à une liste en gardant la liste triée. Il va donc mettre l'élément à la bonne position. Par exemple, si $L = \langle 2, 4, 6 \rangle$ et l'élément à ajouter $E = 5$, l'algorithme va retourner la liste $\langle 2, 4, 5, 6 \rangle$
 - iii. Delete : $\text{Liste} \times \text{Entier} \longrightarrow \text{Liste}$
Cette opération supprime la première occurrence d'un élément E dans une liste
 - iv. Serach : $\text{Liste} \times \text{Entier} \longrightarrow \text{Booléen}$
Cette opération cherche si un élément E existe dans une liste L . Il retourne **vrai** si l'élément existe, sinon il renvoi **faux**.
2. On considère un tableau T contenant n éléments, où chaque élément est une pile (à accès indirect). Chacune de ces n piles peut être éventuellement vide ou peut contenir des éléments classés par ordre croissant d'après leur champ valeur "val" (le sommet d'une pile contient l'élément le plus grand)
 - (a) Donner l'algorithme qui permet de fusionner en une seule pile P tous les éléments contenus dans les n piles, de manière à ce que les éléments de P soit triés par ordre croissant également,
 - (b) Calculer sa complexité en considérant qu'il y a p_1, \dots, p_n éléments respectivement dans les n piles de départ
3. écrire et implémenter un algorithme recevant en entrée une expression arithmétique en postfixe. Si l'expression n'est pas valide on affiche un message indiquant l'erreur. Sinon on calcule et affiche le résultat de l'expression.
4. (Examen 2013/2014) L'objectif de cet exercice est d'implémenter une file avec un tableau (accès direct). Pour cela on considère la structure de données suivante :

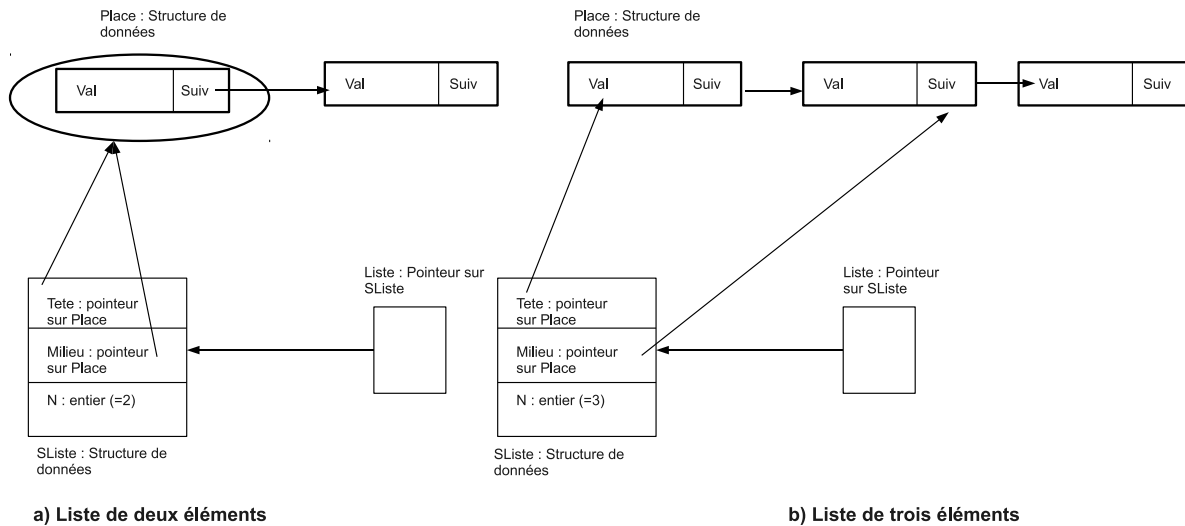


FIGURE 1 – Représentation sémantique du modèle demandé

```

struct File
{
    debut : Entier
    fin : Entier
    compteur : Entier
    Place : tableau de n élément
};

```

La valeur du début indique toujours l'indice du premier élément et celle de fin indique l'indice du dernier élément. La variable compteur donne le nombre d'éléments dans la file.

Dans la figure 4, on trouve un exemple montrant le fonctionnement de cette file.

- Écrire un algorithme vérifiant si la file est vide ou non ?
- Écrire un algorithme vérifiant si la file est pleine ou non ?
- Afin d'enfiler un élément (ajout à la fin), on doit calculer l'indice où on doit le mettre. Donnez la valeur de cet indice en fonction de la variable Fin ?
- Pour défiler un élément (voir la figure), on modifie la valeur de la variable début pour pointer à l'élément qui a été inséré juste avant ?
- Ecrire l'algorithme "Enfiler" qui ajoute un élément à la fin de la file ?
- Ecrire l'algorithme "Retirer" qui retire le premier élément de la file ?

1	2	...	n

Debut = 0
Fin = 0
Compteur = 0

a) La file est vide

1	2	...	n
		12	

Debut = 3
Fin = 3
Compteur = 1

c) Après le retrait de 2 éléments

1	2	...	n
26		12	17

Debut = 3
Fin = 1
Compteur = n - 1

e) Après l'ajout d'un élément

1	2	...	n
7	5	12	

Debut = 1
Fin = 3
Compteur = 3

b) Après l'ajout de 3 éléments

1	2	...	n
		12	17

Debut = 3
Fin = n
Compteur = n - 2

d) Après l'ajout successif des éléments

FIGURE 2 – Exemple de la structure de données demandée