

STRUCTURES DE DONNEES / 2005-2006 / EMD°3

Doc. Non autorisés / 2 Heures

Barème

A = 6+4 ; B= 3 + 5 + 2

Accès rapide dans un fichier L Ô F

A. En vue d'accélérer la recherche par position des articles d'un fichier LÔF (Liste de blocs, sans ordre, format fixe des articles), on utilise une table V_adr contenant les adresses de certains articles. On choisit arbitrairement (par exemple) de mettre dans V_adr[I] l'adresse de l'article de position $K \cdot I$ dans le fichier (K donnée). Une adresse étant le couple (numéro du bloc, déplacement dans le bloc). On considère $K \leq B/2$. B étant la capacité d'un bloc.

1. Ecrire le module qui construit un tel fichier à raison de X articles par bloc ($K \leq X \leq B$) et une telle table à partir de N articles lus. Un article est réduit à sa clé.
2. Donner le module qui recherche l'article de position Pos dans le fichier (Pos est un entier) et retourne l'adresse de l'article (Adr, Depl) avec Adr l'adresse du bloc et Depl le déplacement de l'article dans le bloc) s'il existe. [**Acces_position (Pos , Adr , Depl)**]
Ce module affichera également l'article trouvé.

B. Pour pouvoir insérer de nouveaux articles dans le fichier, on rajoute une table additionnelle V_Ajust de même taille que V_adr contenant les ajustements nécessaires permettant de localiser (rechercher) par position les articles. Lors de la construction, les éléments de V_Ajust sont mis à zéro. A chaque insertion d'un nouveau article à une position donnée Pos dans le fichier correspondant à l'entrée Indice_vadr de V_adr

- si la place est disponible dans le bloc, on rajoute +1 à tous les Ajust[i] tel que $i > \text{Indice_vadr}$ et la composante 'numéro de bloc' de V_adr[i] # Num_bloc, Num_bloc étant l'adresse bloc correspondant à l'entrée Indice_vadr. On ne touche pas à la table V_adr.

- si le bloc est saturé, un nouveau bloc est alloué pour le fichier et la moitié des articles du bloc saturé migre vers le nouveau bloc. Les tables V_adr et V_ajust seront modifiés en conséquence.

3. Réécrire le module de recherche en tenant compte de cette nouvelle structure : le fichier LÔF et les deux tables V_adr et V_Ajust. [**Acces2_position (Pos , Adr , Depl, indice_Vadr)**]

Indice_vadr : indice dans la table V_adr correspondant au couple (Adr, Déplacement)

4. Donner le module d'insertion d'un article à une position Pos donnée dans le fichier.
5. Peut-on garder les mêmes algorithmes si K est quelconque ($B/2 < K \leq B$ ou même $K > B$) ? Justifier.

Exemple :

Prenons comme paramètre $K = 3$ et $B = 8$;

Après insertion des 19 articles suivants : ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's']

On obtient pour $X = B-1$, le fichier suivant :

___ Le fichier bloc par bloc

Bloc n° 1 : a b c d e f g

Bloc n° 2 : h i j k l m n

Bloc n° 3 : o p q r s

V_adr[i,1]	V_adr[i,2]	V_ajust[i]
1	3	0
1	6	0
2	2	0
2	5	0
3	1	0
3	4	0

Après insertion de x à la position 3

___ Le fichier bloc par bloc

Bloc n° 1 : a b x c d e f g

Bloc n° 2 : h i j k l m n

Bloc n° 3 : o p q r s

V_adr[i,1]	V_adr[i,2]	V_ajust[i]
1	3	0
1	6	0
2	2	1
2	5	1
3	1	1
3	4	1

Après insertion de y à la position 14

___ Le fichier bloc par bloc

Bloc n° 1 : a b x c d e f g

Bloc n° 2 : h i j k l y m n

Bloc n° 3 : o p q r s

V_adr[i,1]	V_adr[i,2]	V_ajust[i]
1	3	0
1	6	0
2	2	1
2	5	1
3	1	2
3	4	2

Après insertion de z à la position 9

___ Le fichier bloc par bloc

Bloc n° 1 : a b x c d e f g

Bloc n° 2 : z h i j k

Bloc n° 4 : l y m n

Bloc n° 3 : o p q r s

V_adr[i,1]	V_adr[i,2]	V_ajust[i]
1	3	0
1	6	0
2	2	1
2	5	1
4	3	1
3	1	3
3	4	3