
Fiche TP N°2

Organisation des fichiers

Exercice 1 : Fichier structuré en tableau

Un fichier peut être organisé en un tableau de M blocs. Chaque bloc peut contenir B articles. L'article est réduit à sa clé. Le fichier est trié, c'est à dire :

- À l'intérieur du bloc les articles sont ordonnés,
- Tout article dans le bloc I est inférieur à tout article dans le bloc $I+1$ ($I=1, 2, \dots, M-1$).

Dans un fichier organisé ainsi, on commence par faire un chargement initial de façon à remplir équitablement les blocs.

1. Écrire un programme de la recherche dichotomique.
2. Écrire un programme qui insère un article de clé donnée.

NB. Afin d'éviter les décalages d'articles entre blocs (c'est à dire d'un bloc vers un autre) causés par une insertion, on peut utiliser une zone commune pour ranger les articles qui débordent de certains blocs. La zone commune est un autre fichier composé de N blocs.

Exercice 2 : Fichier organisé en liste linéaire chaînée

Soit un fichier organisé en liste linéaire chaînée de blocs non ordonnée. Un bloc a la structure suivante :

Champ1 : nombre d'articles dans le bloc ;

Champ2 : tableau d'octets contenant les articles ;

Ce tableau est organisé comme suit : Chaque article occupe L octets dont les 3 premiers désignent la longueur.

Champ3 : Lien vers le bloc suivant.

Le bloc d'entête contient l'adresse du premier bloc du fichier.

1. Écrire un programme qui libère (ou supprime physiquement) les K premiers articles de chaque bloc d'un fichier. Si un bloc a un nombre d'articles $< K$, il sera libéré. De plus, chaque bloc libéré est sauvegardé dans le même fichier pour être réutilisé lors d'insertions futures.
2. Écrire un programme qui compacte le fichier de la manière suivante : fusionner le premier bloc avec tous les blocs suivants possibles. Continuer de la même manière à partir du bloc qui suit le dernier bloc fusionné. Fusionner deux blocs $b1$ et $b2$ consiste à déplacer tous les articles de $b2$ vers $b1$ et la libération du bloc $b2$. Chaque bloc libéré est sauvegardé dans le même fichier pour être réutilisé lors d'insertions futures.

Exercice 3 : Indexation pour l'accès multi-critères

Considérons la méthode d'accès suivante permettant d'adresser un fichier à travers un index primaire et éventuellement un ou plusieurs index secondaires :

Soit le fichier F de données est non ordonné et est un ensemble d'articles. Un article possède 4 attributs (Cp, X, Y, Z) où Cp désigne la clé primaire et les autres, les clés secondaires.

Le fichier d'index primaire est un ensemble de couple (Cp, @), où @ désigne l'adresse de l'article et Cp la clé primaire. Une adresse est le couple (numéro du bloc, position de l'article dans le bloc). De plus, les caractéristiques suivantes sont récupérées :

- Nbr_Articles : nombre d'articles du fichier de données,
- Nbr_Blocs : nombre de blocs du fichier de données,
- Ax, Ay, Az désignent les variables destinées à recevoir les adresses des indexes sur X, Y et Z respectivement. Elles sont donc initialisées à Nil,
- Kx, Ky, Kz désignent les nombres de clés secondaires respectifs. Elles sont initialisées à 0.

1. Définir le bloc du fichier de données et la table d'index secondaire.
2. Écrire un programme correspondant à l'opération "Indexer("attribut")" évoquée ci-dessus. Les clés secondaires sont insérées en ordre croissant au fur et à mesure de leur récupération (lecture) du fichier de données.
3. Développer les fonctions suivantes :
 - **Index secondaire (Cs, "attribut")** qui donne l'ensemble des articles ayant Cs comme clé secondaire sur "attribut".
 - **Index secondaire2 (a, b, "attribut1", c, d, "attribut2")** qui donne l'ensemble des articles ayant des clés secondaires dans l'intervalle [a, b] sur attribut1 et des clés secondaires dans l'intervalle [c, d] sur attribut2.

Exercice 4 : Arbres B

On convient de choisir un arbre B comme méthode d'accès à un fichier. Chaque nœud représente alors un bloc sur le disque. Dans chaque nœud et pour toute clé on ajoute un pointeur vers l'information qui se trouve dans un autre fichier structuré sous forme d'un ensemble de blocs contigus. Nous supposons que l'article est réduit à sa clé.

Les programmes demandés :

1. Recherche d'un article de clé donnée.
2. Listage de toutes les clés du fichier en ordre croissant.
3. Insertion d'un article de clé donnée.