

TD6/TP6

Exercice 1 :

On veut gérer un fichier de données formé par N blocs avec une fonction d'hachage 'h' statique utilisant l'algorithme de l'essai linéaire comme méthode de résolution de collisions. Les enregistrements sont de simples entiers. Avec cette méthode, deux types de suppressions sont possibles :

- Suppression physique entraînant éventuellement le parcours de plusieurs blocs en plus de celui contenant la donnée à supprimer
- Suppression logique entraînant une augmentation de l'encombrement mémoire (sur disque)

Afin de limiter les inconvénients des deux méthodes de suppression, on décide de combiner les deux approches comme suit :

Lors de la suppression s'il y a risque de parcourir plusieurs blocs, on réalise alors à la place une suppression logique.

Lors d'une insertion, on essaye de réutiliser d'abord les emplacements des données effacées logiquement et situés sur le chemin parcouru par la recherche

1. Comment est représentée une case vide ? comment être sûr qu'il y'a toujours au moins une case vide dans le fichier ?
2. Donner les caractéristiques du fichier ainsi que la déclaration d'un buffer en spécifiant comment représenter les données effacées logiquement ?
3. Donner l'algorithme permettant de supprimer une donnée x et un autre pour insérer x le plus proche possible de son adresse primaire

On peut utiliser le module ci-dessous, de recherche d'une donnée x, sans l'écrire :

Recherche (entrée x :entier, sorties :trouv :bool, i,j : entiers). Les résultats de la recherche sont :

- Le booléen trouv indiquant si x existe ou non dans le fichier
- I et j représentant l'adresse (numéro de bloc et de déplacement) de la donnée x, si elle existe ou alors, celle du premier emplacement disponible pour l'insérer (si x n'existe pas dans le fichier)

Exercice 2 :

On manipule des enregistrements à taille fixe avec des clés de type entier.

Soit $h(c) = (c \bmod N) + 1$, une fonction de hachage pour accéder à l'enregistrement de clé c dans un fichier vu comme tableau, formé par des blocs ayant une capacité de b enregistrements. En cas de collision sur le bloc de numéro h(c) (c-à-d que le bloc est déjà plein), l'enregistrement de clé c sera inséré en fin de fichier (à partir du bloc N+1). Cette partie du fichier, gérée séquentiellement (et formée par les blocs N+1, N+2, ...), est appelée zone de débordement.

Quand la zone de débordement deviendra trop importante, on effectuera une réorganisation en construisant un nouveau fichier avec une valeur de N plus grande.

- Donner les caractéristiques du fichier.
- Donner le module de recherche d'un enregistrement de clé c.