Examen semestriel de « Structures de Fichiers » (SFSD) – 2CPI – 2014/2015

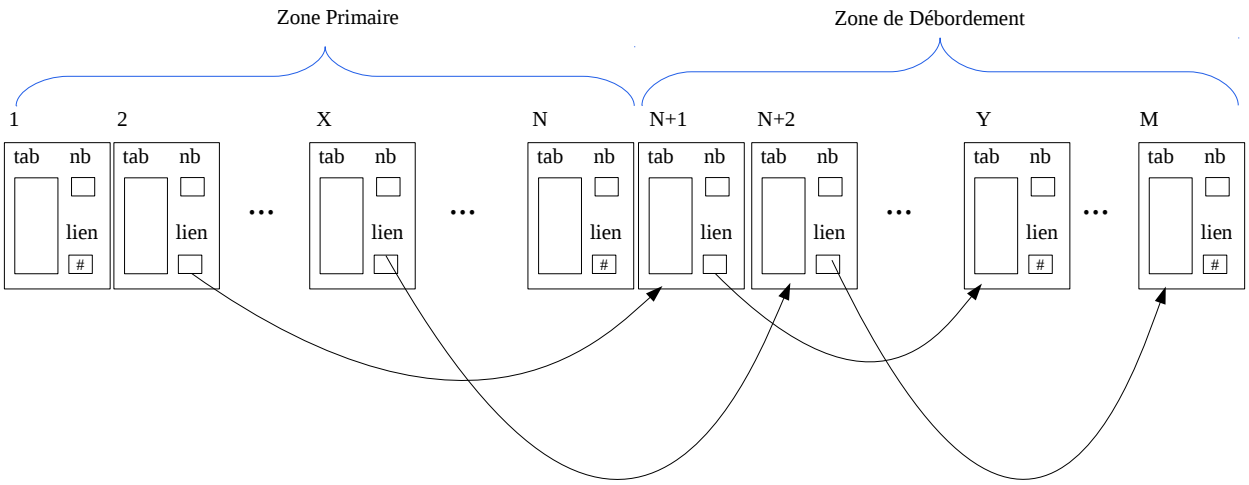
Durée 2h / Documents Interdits / Barème : Problème (4+5+5) , Cours (2+4)

Énoncé du problème :

On définit une structure de fichier ordonné avec gestion d'une zone de débordement. Les enregistrements sont à taille fixe et munis d'une clé sur un type totalement ordonné.
Le fichier, appelé F, est composé de deux parties :

- Une zone primaire ordonnée et formée de N blocs contigus. Cette zone est gérée comme une structure TOF particulière. Sa taille (N) ne peut changer que durant des opérations de réorganisation.
- Une zone de débordement, ordonnée par parties, qui commence à partir du bloc N+1 et s'étend au fur et à mesure que les enregistrements en débordement augmentent. Cette zone est gérée comme un ensemble de structures LOF. Pour chaque bloc X qui déborde dans la zone primaire, est associée une liste en zone de débordement de tête le lien du bloc X. Cette liste de blocs renferme les enregistrements qui n'ont pas pu être insérés dans le bloc X, car devenant plein.

Les blocs peuvent contenir au maximum, b enregistrements.



Lors du chargement initial, on remplit la zone primaire, avec un taux de remplissage fixé au départ, avec des enregistrements donnés en ordre croissant. La zone de débordement sera initialement vide. Le dernier enregistrement de la zone primaire représente une donnée fictive avec comme clé, la valeur infinie (la plus grande valeur du type de la clé).

La localisation d'un enregistrement de clé c, commence par une recherche dichotomique dans la zone primaire et se poursuit, éventuellement, en zone de débordement, séquentiellement sur une des listes de cette zone.

L'insertion d'un enregistrement dans le bloc i (en zone primaire) à la position j, provoque naturellement des décalages intra-bloc. Pour éviter par contre les décalages inter-blocs, dans le cas où le bloc primaire (le bloc i) est déjà plein, on déplace un des enregistrements du bloc i et on l'insère en zone de débordement dans la liste pointée par le champ lien du bloc i si elle existe déjà, sinon elle sera créée et le champ lien mis à jour en conséquence. Pour réaliser cette opération de déplacement d'un enregistrement, lors d'une insertion dans un bloc plein, on effectue les décalages intra-bloc entre les positions j et b-1 uniquement, suivie par l'expulsion de l'avant dernier enregistrement du bloc (position b-1) et son insertion en zone de débordement. De cette manière, le dernier enregistrement de chaque bloc primaire ne pourra jamais être déplacé en zone de

débordement. Cette manière de procéder, nous garantit de pouvoir effectuer des recherches dichotomiques sur la zone primaire pour localiser rapidement un enregistrement de clé donnée.

L'insertion d'un enregistrement dans la zone de débordement suit le schéma général des insertions dans des structures de type LOF. Une fois le bloc de la liste localisé (séquentiellement) l'enregistrement y est inséré. Si le bloc est déjà plein, celui-ci éclate en allouant un nouveau bloc à la fin de la zone de débordement (M+1) et les enregistrements du bloc ayant subit l'éclatement, seront répartis équitablement (moitié/moitié) entre les deux blocs. Le chaînage est bien sûr mis à jour pour garder la structure de liste.

Les suppressions sont effectuées logiquement.
Les caractéristiques du fichier F sont :

N : le numéro du dernier bloc de la zone primaire
M : le numéro du dernier bloc de la zone de débordement
nbI : nombre d'enregistrements insérés

Questions sur le problème : (on peut supposer que le fichier F est déjà ouvert)

P1) Donnez un algorithme efficace 'Localiser(c,i,j)' pour localiser un enregistrement de clé c donnée dans la zone primaire de ce fichier. L'algorithme devrait retourner (dans i) le numéro du bloc primaire ainsi que la position (dans j) où devrait se trouver l'enregistrement (en zone primaire seulement).
Donnez la complexité de votre algorithme.

P2) Donnez un algorithme efficace 'Lister(a,b)' pour effectuer une requête à intervalle.
L'algorithme devrait afficher tous les enregistrements dont la clé $\in$ [a,b].
Quelle est l'influence de la taille des listes de débordement sur les performance de cette opération ?

P3) Donnez l'algorithme 'Reorganiser(taux, nouvNom)' permettant de réorganiser le fichier.
Il s'agira de construire un nouveau fichier de nom 'nouvNom' à l'aide des enregistrements non supprimés logiquement de l'ancien fichier. Le paramètre 'taux' indique le chargement initial voulu pour le nouveau fichier.

Questions de cours :

C1) Comparer les structures de fichiers suivantes :

- Hachage statique et Hachage dynamique.
- B-arbres et B^+-arbres.

C2) Soit les tables relationnelles suivantes, utilisées par un fournisseur de service de téléphonie :
Appels(numAppelant, numAppelé, dateApp, heureApp, duréeApp, coûtApp)

/* chaque ligne de cette table représente les informations concernant un appel :
numAppelant représente la source de l'appel et numAppelé représente la destination */

Abonnés(numTel, nomAb, adrAb)

/* chaque ligne de cette table représente les informations concernant un abonné */

Exprimez les requêtes suivantes en langage relationnel algébrique :

Q1 : Quels sont les abonnés (leurs noms) qui ont fait au moins un appel le '15/04/2015' ?
Q2 : Quels sont les noms des abonnés ayant été appelés par l'abonné de nom 'Ali' entre le '28/10/2013' et le '02/11/2013' ?

On peut comparer les dates avec les opérateurs usuels : <, <=, >, >= , = , <>