Structure de fichiers / EMD n°3 / 98-99 /

Problème : ARBRE DE RECHERCHE BINAIRE COMME INDEX.

Soit la méthode d'index suivante :
-L'index est un arbre de recherche binaire de triplets (Clé, numéro du bloc, déplacement).

- La sauvegarde de l'index consiste à sauvegarder uniquement les triplets le composant en ordre croissant selon la clé dans un fichier structuré en Tableau Ordonné Fixe.

- Le chargement du fichier d'index construit un arbre de recherche binaire le plus équilibré possible. Afin de réaliser ceci, on insère les éléments dans l'arbre selon l'ordre dichotomique de manières externe et interne.

-Le fichier de données est un tableau non ordonné avec format variable des articles.

-Les chevauchements ne sont pas permis.

-Logiquement, un article est composé d'une clé de 4 caractères et d'une information contenant au moins 4 caractères.

-Physiquement l'article est composé d'un octet de présence, suivi de la longueur de l'article sur 3 caractères, suivi de l'article logique. Donc la longueur minimale d'un article est égale à 12. L'octet de présence peut être 'V' (valide), 'E' (effacé) ou 'T' (Trou).

-Un trou de longueur supérieur ou égale à 12 est considéré comme un article.

Algorithmes demandés :

1. Lister le fichier en ordre séquentiel.
2. Lister le fichier en ordre croissant.
3. Sauvegarder l'index.
4. Charger l'index en mémoire.
5. Quel est le nombre maximum d'articles que l'on peut créer avec cette structure de fichier si on dispose de 200K pour l'index avec les occupations mémoires suivantes :
   Pointeur : 2 octets,
   Numéro de bloc : 2 octets,
   Déplacement : 1 octet.

Conseils :
1. Il est plus facile de donner des versions récursives pour 2. et 4.
2. Il est fortement conseillé d'exprimer les algorithmes en Z.

Facilitées :
On utilisera (sans les développer) les modules de conversion :
E(chaîne) : convertit une chaîne en un entier
C(entier) : convertit l'entier en une chaîne de 3 caractères.

et les modules de transfert de caractères :
Copier(C, L, deb) : copier la chaîne C de longueur L à partir de la position Deb dans le buffer.
Extraire(Deb, L, C) : extraire dans C la partie du buffer commençant en Deb et de longueur L.

QUESTIONS

*Initialisation : Définir tous les paramètres nécessaires à l'exploitation du fichier (caractéristiques) et dire comment les initialiser. (2 pts)

*Organisation des N premiers blocs : Donner une organisation possible du bloc 0 contenant les caractéristiques du fichier et une organisation possible des blocs contenant la table d'index (fournir des schémas respectifs). (2pts)

*Fermeture : Ecrire le module qui sauvegarde les caractéristiques et la table d'index. (4 pts)

Afin d'éviter la réorganisation, on décide de récuperer dynamiquement les fragments d'espace devenus inaccessibles ("trous") par des suppressions d'articles. Pour cela on maintient une liste linéaire chainée de "trous" sur le disque.

- Que faut-il mettre dans ces trous ? (1 pt)
- Donner une organisation possible. (2 pts)
- Peut-on récuperer l'espace à 100% ? justifier. (1 pt)

*** Gestion des trous

Ecrire les modules de :

- recherche d'un trou de longueur supérieure ou égale à une longueur L donnée. (3 pts)
- d'insertion d'un trou.(3 pts)

** Opération d'insertion et de suppression :

Paramétrer les modules de gestion des "trous" pour écrire:

- le module de suppression d'un article de clé X donnée.(4 pts)
- le module qui insère un article de clé K donnée et de chaine Y donnée (8 pts)