

LIF4 - TD8

Optimisation

Exercice 1:

Supposons un fichier ordonné avec $r = 30,000$ enregistrements stockés sur un disque avec des blocs de taille $B = 1024$ octets. Les enregistrements sont de taille fixe avec une longueur $R = 100$ octets.

1. Calculer le nombre d'enregistrements par bloc
2. Calculer le nombre de blocs nécessaires pour le fichier

Une recherche binaire (ou dichotomique) dans un fichier ordonné est effectuée de la manière suivante:

- On se donne une variable contenant un intervalle. Tout au long de l'algorithme cet intervalle contient la valeur cherchée si elle est présente dans le fichier.
- Au départ les bornes de l'intervalle sont le premier et le dernier enregistrement du fichier.
- On répète ensuite l'action suivante jusqu'à obtenir un intervalle de taille 1 enregistrement qui contient l'enregistrement cherché s'il est dans le fichier:
 - Diviser l'intervalle en deux parties égales en utilisant l'enregistrement *pivot* situé au milieu des enregistrements "bornes" de l'intervalle courant.
 - Comparer l'enregistrement cherché à *pivot*. S'il est plus petit, changer l'intervalle en prenant le premier sous-intervalle. Sinon changer l'intervalle en prenant le second sous-intervalle.

Donner le nombre d'accès blocs nécessaires pour une recherche binaire dans le fichier ci-dessus.

Exercice 2:

Soit une base de données avec les relations suivantes :

- Employe(Prenom, Nom, NSS, DN, Adresse, Sexe, Salaire, SuperNSS, NumDep)
- Departement(NomD, NumD, NSSResp, DateEntreeResp)
- EmplacementDept(NumD, Emplacement)
- Projet(NomProjet, NProjet, Emplacement, NumD)
- TravailleSur(TNSS, NumProjet, Heures)

Considérons la requête suivante :

```
Q : SELECT Nom
    FROM Employe, TravailleSur, Projet
    WHERE NomProjet='Aquarius' AND NProjet=NumProjet
    AND TNSS=NSS AND DN > '31-DEC-1955'
```

1. Donner en langage naturel le résultat de Q

2. Donner l'arbre de requête optimal pour Q . On traduira d'abord Q en une expression de l'algèbre relationnelle que l'on optimisera ensuite.

```
Q2 : SELECT E1.Nom, Projet.NomProjet
      FROM Employe E1, Employe E2, Departement, Travailleur, Projet
      WHERE E1.SuperNss=E2.Nss AND Projet.NProjet=Travailleur.NumProjet
      AND Travailleur.TNss=E1.Nss AND E2.Nss=Departement.NssResp
      AND E2.Sexe='M' AND E2.Salaire > 10000
      AND Travailleur.Heures > 20
```

Mêmes questions pour Q2.

Exercice 3:

Soit une base de données avec les relations suivantes :

- LIVRE(ISBN, titre, , auteur, éditeur)
- Prêt(NumCarteP, ISBNP, date)
- Lecteur(NumCarte, nom, adresse)

Soit la requête à optimiser: la liste des noms des lecteurs et des titres pour tous les prêts d'avant le 15.3.06.

Donner la requête en SQL puis traduire cette requête en algèbre relationnelle. Donner l'arbre d'optimisation de la requête.