

### 3-a-

Dans la partie 1 du TP , on a déjà utilisé un arbre de recherche binaire a fin d'enregistrer les ID des élèves puisqu'ils sont uniques , et l'arbre facilitera la recherche/verification si l'ID est déjà utilisé ou non .donc on a décidé de la ré-utiliser pour construire notre fichier archive , car ça va diminuer les nombre d'accès disque vers le fichier initial F, et on a procéder comme suit :

- On améliore la structure de l'arbre en ajoutant autres champs : Nom, prenom, supp, année \_scolaire et la moyenne correspondante á l'année \_scolaire .
- On optimise/ ajoute les mises à jour nécessaires dans les fonctions précédemment créées .donc à la fin de la création du fichier élèves ,on aura notre arbre avec tous ses champs remplis , et tout opérations sur le fichier des élèves (insertion , Maj ,suppression ) s'imposera une sur l'arbre (si elle est nécessaire ).
- La création initiale de l'archive va être faite par le parcours de l'arbre (Parcours Inordre ) et le remplissage du fichier archive par les informations des élèves (nom ,prenom, ID)+ l'année scolaire et la moyenne calculée .
- A la fin de chaque année scolaire, on archive des résultats des élèves en 2 étapes globales :
  - i. M-A-J de l'arbre : simultanément avec le chargement du fichier des élèves ,d'où l'arbre va contenir les anciens élèves avec l'année scolaires incrémenté (ou pas si ils sont en 5 eme année) + la nouvelle moyenne calculée . et les nouveaux étudiants des années préparatoire (+100 élèves chaque année).
  - ii. M-A-J archive : en utilisant l'arbre , on insère les étudiants des années préparatoires et on ajoute les informations de l'année dernière pour les anciens.

→ Comme l'arbre est ordonnée selon les identifiants des élèves , alors la clé sera l'identifiant (ID).

→ Le type du modèle choisi : TOF.

3-b- Comme le fichier archive est de type TOF (ordonné ), et il s'agit d'une recherche par intervalle, il faut choisir un index du type non dense.

On a choisi ce type et on a améliorer la structure de cet index en ajoutant le ID min et max par bloc afin de l'utiliser dans les autres opérations (recherche,insertions....).

Par exemple , dans l'insertions ..il suffit juste de faire une recherche dans la table d'index pour trouver la position i, j ... donc 1 seule lecture +écriture sans avoir besoin de la recherche dichotomique dans le fichier TOF.

→ Type index choisi : Index non dense.

### 4-a-

## **le coût de l'archivage :**

On utilise l'arbre pour le remplissage de l'archive ,donc on aura pas besoin du fichier principale .

### **A. Creation de l'archive (année 1 ) :**

- On choisit :
  1. Les blocs de l'archive avec capacité maximale de 25 enreg par bloc.
  2. Taux de chargement : 10 eleves par bloc (pour eviter les decalage cause par les insertions dans les années suivantes
- On a 600 élèves au debut , et 10 eleves par bloc donc initialement il y'a 60 blocs dans l'archive.

Le coût totale = 60 écriture .

### **B. Archivage des années precedents (4 années) :**

- Archivage des résultats des anciens élèves : 1 seul parcours séquentiels du fichier archive , supposons que l'archive ayant N bloc , donc le coût = N lecture + N écriture pour chaque années.
- Insertion des élèves (nouveaux) : insertion avec décalage sans recherche , mais en utilisant l'index de l'archive , et on a 100 nouveaux élèves chaque année .  
Supposons N : nombre de blocs et prenons le pire des cas : tous les blocs sont remplis + décalage de tout le fichiers , le coût :  $100 (N \text{ lecture} + N \text{ écriture})$

Le coût totale :  $101 (N \text{ lecture} + N \text{ écriture})$

D'où le coût d'archivage pour 5 année :  $404 (N \text{ lecture} + N \text{ écriture}) + 600$  écritures.

Dans le cas ou on insère des nouveaux élèves dans le fichier initial, le cout va évidemment augmenter du nombre d'élèves insérés .