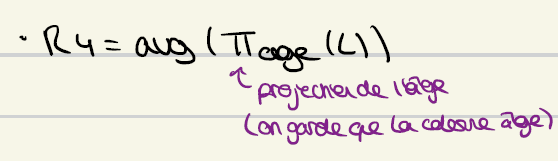
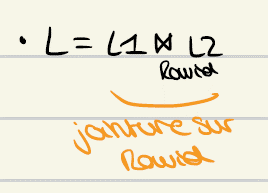
Cheatsheet :

Index :

* La première clé défini le tris de l’index -> important pour le parcourir
* Index couvrant : répondre à une requête sans lire les données, car le résultat de la requete se trouve dans l’index
* Index non plaçant :

Symbole :

Hachage :

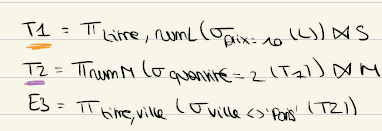
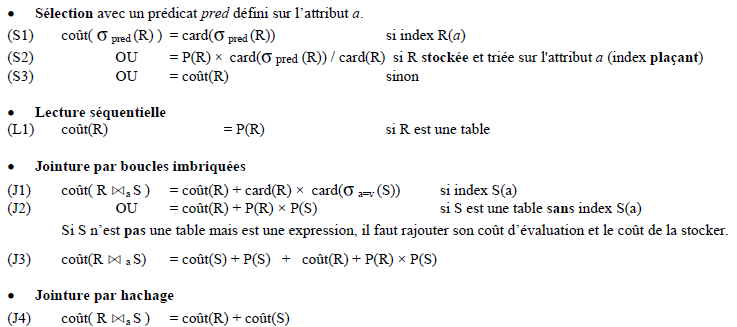
* Un répertoire contient des packets qui contienne des PL valeurs (je pense pointeur)
* Profondeur Globale : = taille du répertoire
* Profondeur Locale : toujours égale à PG au départ, c’est une valeur qu’on vas vite fait bouger et utiliser dans les algos
* Trouver l’ordre de R[A,B] -> Regarder les modulo des valeur à l’intérieur
* Insertion d’une valeur v :
  + On met v dans le packet numéro
  + Si PLi plein :
    - PG = PLi :
      * Doubler le répertoire : R[A,B] -> R[A,B,A,B] (PG = PG + 1)
      * Puis aller au cas suivant :
    - PLi < PG :
      * Créer un nouveau packet Pj
      * Remplacer le deuxième Pi par Pj
        + Si il y a 4 pi : « recopier le remplacement dans le reste du repertoire »
      * Incrémenter les PL de Pi et Pj
      * Repositionner les valeurs de P et la nouvelle valeur v entre Pi et Pj en recalculant les modulos avec la nouvelle valeur de PG
        + Si pi encore plein, refaire l’algo
* Suppression d’une valeur : si un packet i devient vide
  + On peut supprimer un packet vide uniquement si PL = PG, sinon il reste vide
  + Trouver le packet le plus proche ~~qui a le même~~ 
    - Je crois qu’il y a une histoire de suffixe en binaire
  + Le remplacer par ce paquet j
  + Décrémenter le PL du paquet j de 1

Hachage linéaire :

* 3 zones :
  + Une zone de débordement lorsqu’un paquet est plein mais que le taux d’occupation n’est pas dépassé
  + Une zone principale de taille
  + Une zone réservée aux éclatements
    - Quand la taille de cette zone devient plus grand que N on la passe en zone principale : N -> 2N
    - Et on reset p = 0
* Paramètre :
  + N paquet
  + p : prochain paquet à éclater
  + Fonction de hachage tel que
    - On garde en tête et
  + Taux d’occupation :
    - Nombre totale de valeur partout
    - ------------------------------------------------------------------
    - Nombre de cases dans zone principale + extension
    - (on ne compte pas la zone de débordement)
* Accès :
  + Calculer :
  + Si alors lire le paquet
  + Sinon lire le paquet
* Insertion :
  + Tant que le taux d’occupation n’est pas dépassé : insérer et insérer en zone de débordement
  + Dépassement du taux d’occupation
    - Eclater le paquet numéro p
    - Incrémenter p = p + 1
    - Mettre le nouveau paquet dans la zone éclatements
    - Répartir les valeurs entre le paquet éclaté et le nouveau paquet en utilisant la méthode d’accès ci-dessus

Arbre B+

Cout opération :

* Projection = le SELECT final
* Selection = WHERE = filtre
* la jointure sur l’attribue (en particulier l’attribu où S à un index)
* Classiquement sur un arbre linéaire de bas en haut : Jointure → Sélection → Projection
* Expression algèbrique d’un arbre :
  + Arret sur un des traits de l’arbre, écrire avec les en imbriquant tout
  + Ouais tout imbriquer en gros
  + 
* Suivre les formules à la lettre
* 
* Enumérer tous les plans équivalents
  + Penser à la commutativité de la jointure
  + Pas forcément utiliser tous les index disponibles
  + Bon exo 3 TD3 correction Manon
* EXO2 TD3 POTENTIELLEMENT PARTIEL

Rédaction/question cool :

* Exercice 2, question f) 