

## Principales structures d'index

### UE fondements des bases de données - TD8

1. Supposons une mémoire organisée en blocs de taille de 4k, soit 4096 octets. Un pointeur sera codé sur 8 octets. Soit une relation  $FILMS(num.film, titre, date, numstudio)$  triée sur sa clé primaire ; chaque tuple a une taille de  $(8 + 16 + 16 + 8)$  octets. Calculez le nombre de blocs occupé par les objets suivants lorsque la table contient 1.000.000 de tuples, sachant que les blocs ne sont remplis qu'à 80% :
  - (a) La relation
  - (b) Un index primaire dense sur les numéros de films
  - (c) Un index primaire creux sur les numéros de films
2. Combien de chargement de blocs faut-il effectuer au plus pour retrouver une valeur dans la relation sans index ? Avec l'index primaire creux ?
3. Même question avec une deuxième niveau sur cet index.
4. On suppose qu'il y a 80% de valeurs distinctes dans les titres. Calculez le nombre de blocs d'un index sur le titre. N'oubliez pas le "bucket" utilisé pour chaque valeur ; rappelons qu'un bucket est une liste de pointeurs elle même organisée en blocs. On supposera que les blocs stockant les buckets sont pleins.
5. Construire un B-Arbre pour les valeurs suivantes : 49, 17, 21, 16, 34, 42, 81, 39, 41, 27, 1, 5, 83, 6.
  - Considérer  $N=3$ , c'est-à-dire que chaque bloc contient 3 valeurs et 4 pointeurs
  - Insérer les valeurs une à une
  - Faire un nouveau dessin pour chaque nouveau nœud inséré.
6. On se replace dans les conditions de la question 1. Soit un B-arbre sur l'attribut  $NumFilm$  que l'on supposera dense. Quel est le nombre de niveaux dans l'arbre ? Quel est le nombre de blocs à charger pour retrouver une valeur de la clé ? Notez que chaque nœud contient au moins  $\lceil N/2 \rceil$  pointeurs, et donc  $\lceil N/2 \rceil - 1$  valeurs.