

Exercice 1 : Index non-dense

Pour cet exercice, nous considérons l'instance de la relation film visible dans la figure 1. On considère qu'un enregistrement à une taille de 256 octets et une page à une taille de 512 octets (on suppose qu'une page peut contenir deux enregistrements). Un pointeur sur une page (ou RID) occupe 6 octets et la date 4 octets.

MID	TITLE	YEAR	LANGUAGE
2	Ariel	1988	FI
5	Star Wars	1977	EN
3	Shadow in Paradise	1986	FI
7	Dancer in the Dark	2000	EN
80	Nausicaä	1984	JA
71	Billy Elliot	2000	EN
75	Mars Attacks!	1996	EN
62	2001: A Space Odyssey	1968	EN
19	Metropolis	1927	DE
128	Princess Mononoke	1997	JA
103	Taxi Driver	1976	EN
98	Gladiator	2000	EN
391	A Fistful of dollars	1964	IT

Une instance de Film

Question 1 : Combien d'entrées d'index peut contenir une page d'un index de type 2 trié sur les dates ?

Question 2 : Donnez le nombre d E/S (nombre de lecture/écriture sur les pages) nécessaires pour mettre à jour les données et l'index en considérant la figure 1 pour l'ajout de l'entrée (4, Barry Lyndon, 1975, EN) pour un fichier trié (croissant) ? pour un index de type 2 trié (en supposant que le fichier contenant la relation est un tas)?

Question 3 : Répondez à la même question si la relation est contenu dans un fichier de 10 000 pages, et que l'enregistrement sur la première page.

Question 4 : On se replace dans le cas où la relation n'a que 13 enregistrements (voir la figure). À partir de maintenant on va considérer que le fichier de la relation est un tas et qu'une page de l'index de type 2 peut contenir seulement 3 entrées de l'index sur les dates, on nomme cet index trié *A*. Donnez les entrées d'index de l'index *B* non-dense crée sur l'index *A*. Combien de pages sont modifiées pour l'ajout de l'enregistrement (4, Barry Lyndon, 1975, EN) dans l'index *B* ?

Question 5 : Si on utilise cet index, est-il nécessaire que les pages soient consécutivement ordonnées dans l'index *A* (en gardant l'ordre des enregistrements dans les pages)? Combien de pages aura t-on besoin de modifier dans ce dernier cas ?

Question 6 : On réitère l'opération avec un index *C* non-dense sur l'index *B*. Dessinez le nouvel index et déterminez le coût d'une recherche par rapport au nombre d'éléments stockés dans la relation ?

Question 7 : A quoi s'apparente la structure que nous avons crée ? Comment améliorer le coût de l'insertion ?

Exercice 2 : Index B-tree

Nom	Date	Nom	Date
Monteverdi	1589	Faure	1856
Couperin	1668	Beethoven	1770
Rameau	1684	Chopin	1810
Bach	1685	Liszt	1811
Ravel	1875	Bartok	1881
Mozart	1756	Brahms	1833
		Berg	1885
		Bizet	1876

Question 1 : Créer un arbre B+ (d'ordre 2) sur la table pour la clef date par insertions successives dans l'ordre de la table (compter le nombre d'écritures mais aussi de lectures de pages)

Question 2 : Créer un arbre B+ (d'ordre 2) sur la table de la table précédente pour la clef date par bulkloading (remplissage des feuilles à 100%), comparez le nombre d'opérations en lecture et écriture.

Question 3 : Même question avec des feuilles remplis à 50% au maximum.

Question 4 : Proposez deux suppressions sur l'arbre produit à la question n°1 de telle sorte qu'une redistribution des noeuds de même niveau et une fusion de noeuds soient nécessaires

Exercice 3 : Index B-tree, taille et complexité

Supposons qu'un index B+ (entrées de type 2) a été construit sur un fichier non trié contenant 2.104 enregistrements. Le champ correspondant à la clé de recherche de l'index est une chaîne de caractères de 40 octets. La clé de la recherche est une clé de la relation. On suppose que les pointeurs utilisent (au plus) 10 octets. La taille d'une page est de 1000 octets.

Question 1 : Combien de feuilles aura-t-on si tous les noeuds de l'arbre sont pleins ?

Question 2 : Combien d'éléments peut contenir un noeud interne (au maximum) ?

Question 3 : Combien de niveaux a l'arbre si les noeuds et les feuilles sont remplis aux maximum ?

Question 4 : Combien y a-t-il de noeuds à chaque niveau si les noeuds et les feuilles sont remplis aux maximum ?

Question 5 : Supposons que l'on utilise une passe de compression pour diminuer la taille des clés de recherche. La taille d'une clé compressée est de 10 octets. Combien de niveaux aurait alors l'index ?

Question 6 : Combien de niveaux aurait l'index, sans compression, en supposant un taux de remplissage des pages de 60% ?