



## Bases de données avancées

### TD n° 7 : L'organisation physique et les index

#### Exercice 1 :

Les index permettent aux requêtes de s'évaluer plus efficacement. Pourquoi ne pas mettre des index sur tous les attributs ? Trouver le plus grand nombre de raisons possibles et justifier.

#### Exercice 2 :

Une contrainte de clef primaire donne lieu à un index primaire. Peut-on avoir deux index primaires dans une table ?

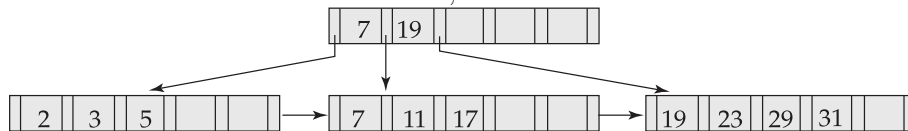
### B+-arbres

#### Exercice 3 :

Les blocs sont de taille 256 octets, les indices se codent sur 32 octets et les pointeurs sur 40 octets. Quelle valeur  $n$  choisit-on pour construire un B<sup>+</sup>-arbre pour une table ?

#### Exercice 4 :

Considérons le B<sup>+</sup>-arbre suivant, avec  $n=5$ .



Faites les opérations suivantes : Ajouter 9, 10, 8, retirer 23, retirer 19.

#### Exercice 5 :

Supposons que la taille des blocs est telle qu'ils peuvent contenir soit trois enregistrements de la table, soit 10 paires de valeurs d'index et de pointeurs plus un éventuel pointeur pour chaîner les blocs entre eux. Donnez en fonction de  $n$  (le nombre total d'enregistrements) le nombre de blocs nécessaires dans le meilleur des cas pour stocker les tables d'index sans overflow lorsque :

1. l'index est dense
2. l'index est éparé
3. l'index est dense, mais aucun des blocs n'est rempli à plus de 80%
4. l'index est éparé, mais aucun des blocs n'est rempli à plus de 80%

**Exercice 6 :**

On reprend les mêmes hypothèses que dans les questions 1. et 2. précédentes en supposant en plus qu'une même clef n'apparaît qu'une seule fois (clef primaire ou attribut unique). On suppose de plus que les blocs de l'index ainsi que ceux des enregistrements sont chaînés entre eux.

1. En quoi le type de l'index (dense ou épars) peut-il changer l'efficacité d'une requête du genre : `select * from T where clef = A;`  
A étant n'importe quelle valeur ?

2. Même question pour une requête du genre :

```
select *  
from T  
where clef is between A and B;
```

A et B étant n'importe quelles valeurs ?