

## Série 2 : CALCUL DE COMPLEXITE

**Exercice 01 :** Soit  $R$  un arbre d'ordre  $m$  et de taille  $n$  et de profondeur (hauteur)  $h$ , montrer que

- $O(h) = O(n)$  dans le cas où  $R$  est un Filiforme
- $O(h) = O(\log_m(n))$  dans le cas où  $R$  est un Arbre équilibré

**Exercice 02 :** Pour chaque type d'arbre (non vide), chaque opération et dans chaque cas (meilleur cas, pire des cas et moyen cas), déterminer une valeur correcte. Cette valeur peut être numérique (0, 1, 2, .....) ou en fonction de  $m$  (ordre),  $n$  (taille) et/ou  $h$  (hauteur ou profondeur).

Type	Opération	Question : donner ..... ?
ABR	Recherche	Le nombre des nœuds parcourus
	Insertion	La profondeur de l'arbre
	Suppression	Le nombre des fils du nœud à supprimer
AVL	Recherche	Le nombre des nœuds parcourus
	Insertion	Le nombre des rotations
	Suppression	Le nombre des rotations
TAS	Recherche	Le nombre des nœuds parcourus
	Insertion	Le nombre des permutations
	Suppression	Le nombre des permutations
AMR	Recherche séquentielle d'une clé	Le nombre des clés parcourues
	Insertion d'une clé dans un nœud non plein	Le nombre des clés après l'insertion
	Suppression d'une clé dans un nœud plein	Le nombre des clés à décaler dans le nœud
B-arbre	Recherche dichotomique d'une clé	Le nombre des clés parcourues
	Insertion dans un nœud plein	Le nombre des éclatements
	Suppression	Le nombre d'emprunt
		Le nombre de fusionnements

**Exercice 03 :** Soit  $R$  un arbre d'ordre  $m$  et de taille  $n$ ,

- Calculer la complexité des algorithmes de recherche, d'insertion et de suppression dans le cas où  $R$  est un :
  - ABR (un arbre binaire de recherche)
  - AVL (un arbre binaire de recherche équilibré)
  - TAS (un arbre binaire équilibré)
  - AMR (un arbre M-aire de recherche)
  - B-Arbre (un arbre M-aire de recherche équilibré)
- Déduire la complexité des algorithmes de tri par ABR, par AVL et par TAS.