

Exercice 1 (5 pts) : choisissez la bonne réponse pour chacune des questions de 1 à 5

- Combien d'arbres binaires de 3 éléments est possible (On différencie seulement selon la topologie, pas les valeurs) ?
 a. 4
 b. 6
 c. 5
 d. 7
- La recherche d'un élément dans une liste chaînée est un algorithme :
 a. Linéaire
 b. Quadratique
 c. Quasi-linéaire
 d. Logarithmique
- Le tri par fusion est plus efficace que le tri par sélection car :
 a. Il consomme plus de mémoire
 b. Il s'exécute dans moins de temps
 c. Il utilise le principe de diviser pour régner
 d. Il est logarithmique dans le pire de cas
- La recherche d'un élément dans arbre binaire nécessite au pire de cas de parcourir
 a. La moitié de l'arbre
 b. Tous les éléments de l'arbre
 c. 2 tiers des éléments
 d. Aucune réponse n'est correcte
- Les files et les piles peuvent être implémentées à l'aide des tableaux
 a. Oui
 b. Non

Exercice 2 (6 pts)

Soit A un arbre binaire dont les éléments sont des caractères de l'alphabet anglais. Son parcours en préfixe donne le mot **SUPNUM** et l'infixe donne le mot **PUNSUM**.

- Dessiner l'arbre A
- Donner l'arbre binaire de recherche B résultant de l'ajout successif des éléments de A ordonnée selon le parcours préfixe.
- Donner l'arbre C résultant de l'ajout successif des éléments K, Z, N, R à l'arbre B
- Donner l'arbre D résultant de la suppression successive des éléments K, P de l'arbre C

Exercice 3 (5 pts)

- Ecrire un algorithme (ou fonction c++) qui prend comme paramètre un arbre binaire des entiers et retourne la somme de ces éléments
- Ecrire un algorithme (ou fonction c++) qui retourne la fréquence (nombre d'occurrences) d'un élément dans arbre binaire de recherche

Exercice 4 (4 pts)

Soit le tableau T dont les éléments sont : 5, 6, 1, 4, 20, 30, 10

- Trier ces éléments en appliquant l'algorithme de tri par sélection (on montre les résultats intermédiaires)
- Même question pour le tri par fusion