

Examen de Rattrapage : Algorithmique et Complexité
(Durée : 2h00)

Questions : (6 pts)

Choisir **la** bonne réponse (une seule) :

1. La complexité temporelle de $O(\log n)$ signifie que :
 - A. L'algorithme effectue une opération logarithmique en fonction de la taille de l'entrée.
 - B. L'implémentation de l'algorithme prend $\log(n)$ unités de temps.
 - C. L'algorithme effectue un nombre d'opérations proportionnel au logarithme de la taille de l'entrée.
2. Dans un arbre binaire, le nombre maximum de nœuds à un niveau « k » (la racine de l'arbre ayant le niveau 0) ?
 - A. 2^k .
 - B. $2 \times k$.
 - C. $2 + k$.
3. L'algorithme de recherche dichotomique a une complexité temporelle de :
 - A. $O(n)$
 - B. $O(\log n)$
 - C. $O(n^2)$
4. Dans un arbre binaire de recherche, quel parcours permet d'obtenir les éléments dans l'ordre croissant ?
 - A. Parcours préfixe.
 - B. Parcours infixe.
 - C. Parcours postfixe.
5. La hauteur d'un arbre binaire de recherche équilibré contenant « n » nœuds est approximativement :
 - A. n.
 - B. $\log n$.
 - C. $n (\log n)$.
6. Le tri par sélection fonctionne en :
 - A. Divisant le tableau en deux parties, triée et non triée, et en déplaçant à chaque fois le plus petit élément de la partie non triée vers la partie triée.
 - B. Divisant le tableau en deux parties d'une même taille et en les triant récursivement.
 - C. Parcourant le tableau et échangeant les éléments adjacents qui ne sont pas en ordre.

Exercice 1 : (6 pts)

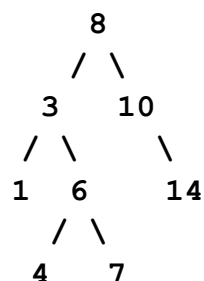
Soit la fonction suivante :

```
int doingSomething(int *T, int n) {  
    // Supposons que le vecteur T contient au moins deux éléments  
    int min = abs(T[0] - T[1]);  
    for (int i = 0; i < n; i++) {  
        for (int j = n - 1; j > i; j--) {  
            int diff = abs(T[i] - T[j]);  
            if (min > diff && diff > 0) {  
                min = diff;  
            }  
        }  
    }  
    return min;  
}
```

- Que fait cette fonction ?
- Quelle est sa complexité temporelle ?
- Expliquer comment peut-on utiliser les algorithmes de tri afin d'améliorer la complexité de cette fonction ?
- Ecrire le code correspondant à la version améliorée.

Exercice 2 : (8 pts)

Soit l'arbre binaire suivant :



1. Donner le parcours préfixe, postfixe, infixe et le parcours en largeur de l'arbre binaire ci-dessus.
2. L'arbre est-il un arbre binaire de recherche ? Justifier.
3. Ecrire une fonction permettant d'afficher les éléments d'un arbre binaire dans l'ordre infixe.