

Exercice 1 (3 points)

```

int recherche (float tab[], int n, float elem)
{
    int i;
    i=0;
    while ((i<n) && (tab[i] !=elem))
        i=i+1;
    if (i==n) i=-1;
    return i;
}

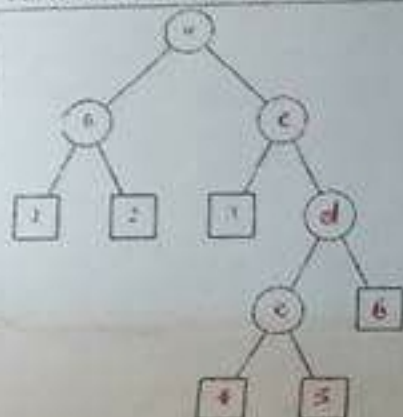
```

Question :

Calculer la complexité de cette fonction.

Exercice 2 (6 points)

Partie 1 : Soit l'arbre suivant :



Questions :

- Donnez le résultat de parcours de cet arbre.
- Quelle est la Taille et la hauteur de cet arbre ?

Partie 2 :

- A partir d'un arbre binaire vide, ajoutez successivement les valeurs suivantes : 22, 09, 41, 39, 27, 06, 11, 45, 59, 18, 36.
- A partir de l'arbre final de la question (a), supprimez successivement les valeurs 59 et 11.
- Supprimez maintenant les valeurs 22 et 41. Quelles valeurs choisissez-vous pour les remplacer ?

Exercice 3 (5 points)

Écrire une fonction récursive `chiffre (n, k)` qui permet de retourner le k-ième chiffre à partir de la droite de `n`. Par exemple Le 3ième chiffre à partir de la droite de 8724 est 7. Le 5ième chiffre à partir de la droite de 21327 est 2

Exercice 4 (6 points)

- Écrire une fonction `swap` qui a comme paramètres deux pointeurs vers des entiers et qui échange le contenu des deux entiers pointés.
- Tester cette fonction en écrivant un programme qui échange le contenu de deux entiers `a` et `b` en appelant cette fonction.