

Exercice 1

Examiner le comportement asymptotique des fonctions suivantes :

1. $f(n) = n^6 + 3n$ | 3. $f(n) = 2^n + 12$

2. $f(n) = 3^n + 2^n$ | 4. $f(n) = n^n + n$ | 5. $f(n) = n + \sqrt{n}$

Exercice 2

Soit deux tableaux T1 et T2 , de différentes tailles, contenant des entiers:

1. Ecrire un algorithme qui calcule la somme des éléments d'un tableau. Compter son nombre d'instructions et déterminer sa complexité.
2. Ecrire un algorithme qui met dans un troisième tableau T3 les éléments pairs de T1 et T2. Déterminer le nombre d'instructions et sa complexité.
3. Ecrire un algorithme qui calcule l'intersection entre T1 et T2, compter le nombre d'instructions et déterminer la complexité.

Exercice 3 (à faire en TP)

Soit l'algorithme suivant

```
int ALGO(int A[n]){
    bool d = false;
    for ( int i = 0; i < n; ++i ) {
        for ( int j = 0; j < n; ++j ) {
            if ( (i != j) && (A[i] == A[j]) ) {
                d = true;
                break;
            }
        }
        if ( d ) {
            break;
        }
    }
}
```

Q1 > Essayer de le dérouler pour le tableau $A=\{1,6,9,3,4,1,2\}$, de même pour le tableau $A=\{1,4,6,9\}$.

Q2 > Compter le nombre d'instructions faites par l'algorithme et déterminer sa complexité.

Exercice 4 (à faire en TP) : soit l'exercice décrit ci-dessous. Que fait cet algorithme et déterminer

```
1 int dichotomie(int *t, int n, int x) {
2     int a,b,mid;
3     a = 0;
4     b = n;
5     while(a <= b) {
6         mid = (b+a)/2;
7         if (t[mid] == x) return 1;
8         if (t[mid] < x) a = mid + 1;
9         else b = mid - 1;
10    }
11    return 0;}
```

sa complexité.