Exercices complexité algorithmique

Exercice 1 : Donner la complexité des fonctions suivantes.

```
# 1

def premier_element(liste):
    return liste[0]
```

```
# 2

def somme_liste(liste):
    s = 0
    for x in liste:
        s = s + x
    return s
```

```
# 3

def toutes_les_paires(liste):
    for i in liste:
        for j in liste:
            print(i, j)
```

Exercice 2 : Compter le nombre d'itérations puis donner la complexité des fonctions suivantes.

```
# 4

def maximum(liste):
    m = liste[0]
    for x in liste:
        if x > m:
            m = x
    return m
```

```
# 5

def compte_pairs(liste):
    compteur = 0
    for x in liste:
        if x % 2 == 0:
            compteur += 1
    return compteur
```

```
# 6

def affiche_moitie(liste):
    n = len(liste)
    for i in range(n//2):
        print(liste[i])
```

```
# 7

def double_parcours(liste):
    for x in liste:
        print(x)

    for x in liste:
        print(x*2)
```

```
# 8

def triangle(n):
    for i in range(n):
        for j in range(i):
            print("*")
```

```
# 9

def recherche_mat(mat, val):
    for ligne in mat:
        for e in ligne:
            if e == val:
                return True
    return False
```

```
# 10

def mystere(n):
    i = 1
    while i < n:
        i = i * 2
        print(i)
```

```
# 11

def comparaison(l):
    n = len(l)
    for i in range(n):
        for j in range(i+1, n):
            print(l[i], l[j])
```

```
# 12

def mystere(n):
    for i in range(n):
        for j in range(10):
            print(i, j)
```

```
# 13

def algo2(n):
    for i in range(n):
        print(i)

    for j in range(n):
        for k in range(n):
            print(j, k)
```

```
# 14

def algo6(n):
    for i in range(n):
        for j in range(n):
            for k in range(n):
                print(i, j, k)
```

```
# 15

def algo7(n):
    for i in range(n):
        for j in range(i, n):
            print(i, j)
```

Exercice 3 : Pour chaque fonction, donner la complexité dans :

- le **meilleur cas**
- le **pire cas**
- le **cas moyen**

```
# 16

def recherche(liste, valeur):
    for x in liste:
        if x == valeur:
            return True
    return False
```

```
# 17

def tri_selection(tab):
    n = len(tab)
    for i in range(n):
        min_i = i
        for j in range(i+1, n):
            if tab[j] < tab[min_i]:
                min_i = j
        tab[i], tab[min_i] = tab[min_i], tab[i]
```