

---

IUT de Vélizy - UVSQ  
Développement Efficace

# 1. Complexité algorithmique - TD

---

## 1. Exercices sur feuille

### 1.1. Domination Asymptotique

1. Montrez que  $n + 5 \in O(n)$
2. Montrez que  $n^2 + n \in O(n^3)$
3. Montrez que  $n^3 \notin O(n)$
4. Montrez que  $n^{10} \in O(2^n)$
5. Montrez que  $2^n \in O(3^n)$
6. Montrez que  $n \notin O(\log n)$
7. Montrez que si  $P(n)$  est un polynôme de degré  $k$  alors  $P(n) \in O(n^k)$
8. Quelles sont les fonctions  $f$  telles que  $f \in O(1)$ ? Démontrez.

### 1.2. Evaluation de complexité

Donnez un ordre de grandeur ( "grand O" le plus fin possible) de la complexité algorithmique (en temps, pire des cas) des codes suivants en fonction de leurs paramètres. On considère les affectations, les affichages et les opérations arithmétiques comme étant du temps constant.

#### 1.2.1

```
pour i de 1 à n:  
    écrire i
```

#### 1.2.2

```
i = n  
tant que i > 2:  
    i = i - 1
```

#### 1.2.3

```
pour i de 1 à n:  
    pour j de 1 à n:  
        écrire (i+j)
```

**1.2.4**

```
pour i de 1 à n:
  pour j de i à n:
    écrire (i+j)
```

**1.2.5**

```
pour i de 1 à n:
  j = 0
  tant que j < i:
    j += 1
```

**1.2.6**

```
pour i de 1 à n:
  j = 0
  tant que j < i*i:
    j += 1
```

**1.2.7**

```
pour i de 5 à n:
  pour j de i-5 à i+5
    écrire (i+j)
  j += 1
```

**1.2.8**

```
tant que n > 0:
  n = n // 2
```

**1.2.9**

```
tant que n > 0:
  n = n // 10
```

**1.2.10**

```
pour chaque choix de a, b et c entre 1 et n :
  écrire a + b + c
```

**1.2.11**

```
pour chaque choix de a, b entre 1 et n tels que a < b:
  écrire a + b
```

**1.2.12**

```
c = 0
pour chaque sous-ensemble de {1,2,...,n}:
    c += 1
```

**1.2.13**

```
l un liste
pour chaque permutation l' de l :
    écrire l'
```

**1.2.14**

Remarque : une sous-liste est obtenue en prenant des éléments de l, dans l'ordre, pas forcément voisins dans la liste.

```
l un liste
pour chaque sous-liste l' de l :
    écrire l'
```

**1.2.15**

Remarque : un facteur d'un mot u est un sous-mot qui apparaît dans u en un seul bloc

```
u une chaîne de longueur n
pour chaque facteur v de u :
    écrire v
```