Introduction à la Complexité Algorithmique

TD n°1

EPITA Cyber 2
2025-2026

Exercice 1 :
Estimez le nombre d'opérations élémentaires dans le pire de cas pour les quatre algorithmes suivant :

A)

```
1 int algoA(int n){
2     int num1, num2, num3, num4;
3     num1 = 2*n + 15;
4     num2 = n + 10;
5     if( (n-10) > 0){
6         return num1 + num2;
7     }
8     return 0;
9 }
```

Nombre d'opérations de algoA dans le pire cas :

- l.2 : 4 déclarations,
- l.3 : 1 multiplication, 1 addition, et 1 affectation,
- l.4 : 1 addition et 1 affectation,
- l.5 : 1 addition, 1 test,
- l.6 : 1 addition et 1 instruction return,
- l.8 : 1 instruction return (qui ne correspond pas au pire cas).

⇒ Soit un nombre total d'opération dans **algoA** de 13 opérations.

B)

```
1  int algoB(int n){
2      if ( n==0 ){
3          return 10;
4      } else if ( n==1 ){
5          return 11;
6      } else if ( n==2 ){
7          return 12;
8      } else if ( n==3 ){
9          return 13;
10     } else {
11         return 0;
12     }
13 }
```

Nombre d'opérations de algoB dans le pire cas :

- l.2 : 1 test,
- l.3 : 1 instruction return (qui ne correspond pas au pire cas),
- l.4 : 1 test,
- l.5 : 1 instruction return (qui ne correspond pas au pire cas),
- l.6 : 1 test,
- l.7 : 1 instruction return (qui ne correspond pas au pire cas),
- l.8 : 1 test,
- l.9 : 1 instruction return (qui peut correspondre au pire cas),
- l.11 : 1 instruction return (qui peut correspondre au pire cas),

$\Rightarrow$ Soit un nombre total d'opération dans **algoB** de 5 opérations.

C)

```
1  int algoC(int n){
2      int cpt=0;
3      for(int i=0; i<n; i++){
4          if( i%2==0 ){
5              for(int j=0; j<n; j++){
6                  cpt++;
7              }
8          }
9      }
10     return cpt;
11 }
```

Nombre d'opérations de algoC dans le pire cas :

- l.5 : la boucle for sur j itère de $j = 0$ à $j = n - 1$,
  → à chaque itération de la boucle for sur j, on compte 3 opérations :
  - l.5 : un test booléen et une incrémentation du compteur j,
  - l.6 : une incrémentation,

  → Pour la boucle for sur j, on a : $\sum_{j=0}^{n-1} 3 = 3 * \sum_{j=0}^{n-1}(1) = 3n$ opérations.
- l.3 : la boucle for sur i itère de $i = 0$ à $i = n - 1$,
  → à chaque itération de la boucle for sur i, on compte :
  - l.3 : un test booléen et une incrémentation du compteur i,
  - l.4 : un modulo, un test,
  - l.5 : une déclaration et une affectation du compteur j,
  - l.5 : $3n$ opérations provenant de la boucle sur j,

  → Pour la boucle for sur i, on compte un nombre d'opérations à :

$$\begin{aligned}\sum_{i=0}^{n-1} 3n + 6 &= \sum_{i=0}^{n-1} 3n + \sum_{i=0}^{n-1} 6 \\ &= 3n \sum_{i=0}^{n-1}(1) + 6 \sum_{i=0}^{n-1}(1) \\ &= 3n^2 + 6n\end{aligned}$$

- l.2 : une déclaration et une affectation,
- l.3 : une déclaration et une affectation du compteur i,
- l.10 : une instruction return.

⇒ Soit un nombre total d'opération dans **algoC** de $3n^2 + 6n + 5$ opérations.

Exercice 2 :
Considérez l'algorithme suivant :

```
1 int algoE(int tab[], int n){
2     for(int i=0; i<n-1; i++){
3         if( tab[i]>tab[i+1] ){
4             return 0;
5             }
6     }
7     return 1;
8 }
```

1. Que fait l'algorithme "algoE" ?
2. Donner deux instances de même taille, correspondant à nombre d'opérations différentes.
3. Donner deux instances de taille 10, une correspondante au pire cas et une au meilleur cas.
4. Analysez la complexité de l'algorithme "algoE", c'est-à-dire, estimez le nombre d'opérations élémentaires exécutées par "algoE", dans le pire cas, en fonction de la taille de l'instance.

1. L'algorithme "algoE" retourne 1 si le tableau est trié dans l'ordre croissant et 0 sinon.
2. Instance A = ({1, 2, 3}, 3) et instance B = ({3, 2, 1}, 3).
   Pour l'instance A, l'algorithme parcourt entièrement le tableau, tandis qu'il s'arrête à la première comparaison pour l'instance B.
3. Instance pire cas = ({1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, 10) et instance meilleur cas = ({9, 2, 3, 4, 5, 6, 7, 8, 9, 10}, 10).
   Pour l'instance pire cas, l'algorithme parcourt entièrement le tableau, tandis qu'il s'arrête à la première comparaison pour l'instance meilleur cas.
4. Nombre d'opérations de algoC dans le pire cas :

   - l.2 : la boucle for sur i itère de $i = 0$ à $i = n - 2$,
     $\rightarrow$ à chaque itération de la boucle for sur i, on compte 5 opérations :
     — l.2 : une soustraction, un test booléen, et une incrémentation du compteur j,
     — l.3 : une addition et un test,
     — l.4 : une instruction return (qui ne correspond pas au pire cas),
     $\rightarrow$ Pour la boucle for sur i, on a : $\sum_{i=0}^{n-2} 5 = 5 * \sum_{i=0}^{n-2}(1) = 5(n-1)$ opérations.
   - l.2 : une déclaration et une affectation du compteur i,
   - l.7 : une instruction return.

   $\Rightarrow$ Soit un nombre total d'opération dans **algoE** de $5(n-1)+3$ opérations.