

introduction_algorithmique

April 8, 2021

1 Qu'est-ce qu'un algorithme ?

1.1 Analogie

Une recette de cuisine est un algorithme :

1. Ajouter 200mL de lait
2. Ajouter 100g de farine
3. Mélanger . . .
4. Tant que la pâte n'est pas levée: Laisser au four à 200°C
5. etc...

1.2 Définition plus rigoureuse

Un **ALGORITHME** est la suite d'**INSTRUCTIONS ELEMENTAIRES** permettant d'effectuer une **TACHE SOUHAITEE** en un **TEMPS FINI**.

Les **instructions élémentaires** sont celles directement liées à ce que la machine sait faire car elle possède les **structures matérielles** permettant de les réaliser (par exemple dans son **Unité arithmétique et logique**) :

- Opérations mathématiques ou logiques, affectation ($3*5$, `a and b`, `a = 3`)
- Tests (`if`)
- Boucles (`for`, `while`)

J'insiste :

1. Ecrire un algorithme consiste à écrire (en français !!) de manière claire et précise la suite d'**instructions élémentaires** (et uniquement d'instructions élémentaires !!) permettant de réaliser la tâche souhaitée.
2. Le seul matériel utile pour écrire un algorithme c'est du papier et un crayon. Surtout pas d'ordinateur : Ecrire un algorithme ne consiste pas à écrire du code python !!
3. Une fois l'algorithme écrit, **nimporte qui** sans connaissance particulière, peut réaliser la tâche souhaitée en suivant l'algorithme
4. Programmer c'est traduire un algorithme dans un langage informatique (par exemple en python)

1.3 Méthodologie du programmeur

“Je dois écrire un programme qui doit réaliser une tâche, pour cela je fais **dans l'ordre**” :

1. Je prends le temps de bien comprendre la tâche demandée

2. je conçois l'algorithme (sans ordinateur !!)
3. je "traduis" cet algorithme dans un langage informatique (python, javascript etc...)
4. je teste mon programme
5. je corrige les bugs

La grosse erreur que les élèves font souvent, c'est de se jeter sur l'ordinateur pour écrire du code python sans passer par l'étape de réflexion avec un papier et un crayon. Bref, de passer directement à l'étape 3 et de se trouver "curieusement" bloqué...

2 Astuces courantes utilisées dans les algorithmes

... Quelques méthodes "basiques" couramment utilisées dans les algorithmes

2.1 Parcourir les éléments VS parcourir les indices

- Parcourir uniquement les indices quand vous avez besoin d'accéder aux indices dans l'algorithme
- Il vaut mieux parcourir les éléments plutôt que les indices (structure plus efficace)

```
[2]: # PARCOURS DES ÉLÉMENTS DE LA LISTE
liste = ['a','b','c','d']
for lettre in liste:
    print(lettre)
```

a
b
c
d

```
[5]: # PARCOURS DES INDICES DE LA LISTE
liste = ['a','b','c','d']

for i in range(len(liste)): # rappel : range(len(liste)) est équivalent à
    ↪ [0,1,2,3]
    print(i, liste[i])
```

0 a
1 b
2 c
3 d

2.2 Les compteurs

Un compteur est une variable (initialisée en dehors de la boucle, souvent à 0) et qui est incrémentée de 1 à chaque passage dans une boucle

```
[10]: compteur = 0 # Initialisation

for lettre in "python":
```

```
compteur = compteur + 1 #Incréméntation
print(compteur)

print("le nombre de caractères dans python est de :", compteur)
```

```
1
2
3
4
5
6
le nombre de caractères dans python est de : 6
```

2.3 Les accumulateurs

Un accumulateur est semblable à un compteur mais qui est incrémenté d'une valeur différente de 1 à chaque passage dans une boucle

```
[6]: accumulateur = 0 # Initialisation
    for i in range(6):
        accumulateur = accumulateur + i #Incréméntation
        print(accumulateur)

    print("la somme des entiers strictement inférieurs à 6 est de :", accumulateur)
```

```
0
1
3
6
10
15
la somme des entiers strictement inférieurs à 6 est de : 15
```