

# Notion d'algorithmique

## I Qu'est-ce que l'algorithmique

### Définition

L'**algorithmique** est la science qui étudie la conception, l'analyse et l'optimisation des **algorithmes**. Un algorithme, c'est comme une recette de cuisine : une suite d'étapes précises pour résoudre un problème ou accomplir une tâche.

L'algorithmique est au cœur de l'informatique, mais aussi de nombreux domaines : mathématiques, biologie, économie, intelligence artificielle, etc. Elle permet de :

- **Résoudre des problèmes** de manière systématique et efficace.
- **Automatiser des tâches** répétitives ou complexes.
- **Optimiser** l'utilisation des ressources (temps, mémoire, énergie).

## II Que permet l'algorithmique ?

L'étude de l'algorithmique permet de répondre à des questions essentielles :

- **L'algorithme fonctionne-t-il toujours ?** (termine-t-il ?)
- **Le résultat est-il correct ?** (fournit-il la bonne réponse ?)
- **Est-il rapide et efficace ?** (temps d'exécution, consommation mémoire)
- **Peut-on faire mieux ?** (existe-t-il un algorithme plus performant ?)
- **Modéliser des problèmes** réels (recherche d'itinéraire, compression de données, etc.).

## III Qu'est-ce qu'un algorithme ?

Un **algorithme** est une suite **finie** et **non ambiguë** d'instructions permettant de résoudre un problème ou une classe de problèmes.

[Donald Knuth](#) dans un ouvrage nommé "The Art of Computer Programming" donne en 1968 cinq caractéristiques fondamentales aux algorithmes :

- **Finitude** : L'algorithme doit se terminer après un nombre fini d'étapes.
- **Précision** : Chaque étape doit être définie sans ambiguïté.
- **Entrées** : Données nécessaires au démarrage de l'algorithme.
- **Sorties** : Résultats produits par l'algorithme.
- **Rendement** : Les opérations doivent être réalisables en pratique (même à la main).

Les algorithmes existaient bien avant l'informatique. Par exemple : [l'algorithme d'Euclide](#) (III<sup>e</sup> siècle av. J.-C.) permet de calculer le plus grand commun diviseur (PGCD) de deux nombres. Le mot algorithme vient du nom du savant arabe [Al-Khwârizmî](#) (IX<sup>e</sup> siècle).

### Remarque :

Un algorithme peut être comparé à une recette de cuisine. En suivant cette comparaison, il ne s'occupe pas de savoir si l'on a les ingrédients, les ustensils comme

un four et la quantité d'énergie pour celui-ci. Par contre, son étude permet d'estimer les ressources nécessaires et le temps qu'il va prendre. De plus, il est indépendant du langage utilisé, c'est pourquoi il est souvent écrit en pseudo code qui est un intermédiaire entre un langage informatique et le langage courant.

## VI Notion de spécification

Un algorithme se compose de trois phases :

1. L'entrée des données : la nature des données d'entrée
2. Le traitement des données
3. La sortie des données : quelles sont les données en sorties et à quoi elles correspondent.

On définit ainsi :

### Définitions

- **Préconditions** : La nature des données d'entrée pour que l'algorithme fonctionne. Par exemple des nombres non nuls.
- **Postconditions** : une postcondition est une propriété que doit être vraie en sortie de l'algorithme.

**Remarque** : Bien documenter une fonction est un travail de spécification, qui peut être complété par l'ajout de la vérification des préconditions au début de la fonction et par un jeu de test pour vérifier les postconditions.

### Exemple

La fonction valeur absolue prend en entrée un nombre et retourne un nombre positif égal au nombre en entrée s'il est positif ou à son opposé sinon. La précondition est de prendre un nombre en entrée. La postcondition est que ce nombre soit positif et corresponde à la valeur absolue du nombre en entrée.

En python cela donne :

```
def val_abs(n):
    """
    n: un nombre
    Retourne la valeur absolue de n
    """
    assert isinstance(n, int) or isinstance(n, float), "n doit être un
    ↪ nombre"
    if n >= 0:
        return n
    else:
        return -n
# jeu de tests
assert val_abs(10) == 10
assert val_abs(0) == 0
assert val_abs(-10) == 10
```