



Programmation algorithmique

Leçon 1 Introduction à l'algorithmie



(Wikipedia)

Muhammad Ibn Mūsā **al-Khuwārizmī**, Perse (Bagdad), 780-850.
Mathématicien, astronome, géographe et auteur.

Plan de leçon

Qu'est-ce qu'un algorithme

Le pseudo-code

Exemples

Exactitude et efficacité (survol)

Les opérations de base

Les limites du pseudo-code

Qu'est-ce qu'un algorithme?

Un algorithme est une suite finie et non ambiguë d'opérations ou d'instructions permettant de résoudre un problème ou d'obtenir un résultat

Wikipedia

Une suite finie d'instructions basiques

Opérant sur une entrée

Produisant une sortie

Dans un temps fini

Knuth

Qu'est-ce qu'un algorithme?

Un algorithme est une suite finie et non ambiguë d'opérations ou d'instructions permettant de résoudre un problème ou d'obtenir un résultat

Wikipedia

Une suite finie d'instructions basiques

Opérant sur une entrée

Produisant une sortie

Knuth

Dans un temps fini

Une recette de biscuits

Une procédure de triage

Une méthode de calcul

Exemple - L'ajustement du prix du loyer.

<https://www.rdl.gouv.qc.ca/fr/calcul-pour-la-fixation-de-loyer/outil-de-calcul>

Ajustement de loyer du logement ⓘ

Loyer mensuel du logement
(avant augmentation)

Reporter le pourcentage F

Multiplié par F

Montant de la case G

Total

Après arrondissement

Exemple - La valeur absolue

Entrée : Un nombre réel x

Sortie : Un nombre réel x

Les instructions:

```
Si  $x < 0$ ,  
Alors  $x \leftarrow x * (-1)$   
Retourner  $x$ 
```

Exemple - Le pgcd (encore?!)

Entrée : Deux nombres naturels x et y

Sortie : Un nombre naturel **pgcd**

Les instructions:

$i \leftarrow 1$

$\text{pgcd} \leftarrow 1$

Tant que $i \leq x$

Faire

 Si i divise x et y sans reste

 Alors $\text{pgcd} \leftarrow i$

$i \leftarrow i + 1$

Retourner pgcd

Exemple - La racine carrée (méthode babylonienne)

Entrée : Un nombre réel x

Sortie : Un nombre réel r

Les instructions :

$r \leftarrow 10$

Tant que $r*r$ est loin de x ,

Faire $r \leftarrow$ la moyenne entre r et x/r

Retourner r

Exemple - La racine carrée (méthode babylonienne)

Entrée : Un nombre réel x

Sortie : Un nombre réel r

Les étapes (méthode babylonienne):

$r \leftarrow 10$

Tant que la valeur absolue de $r*r - x$ est plus grande que 0.0001 ,

Faire $r \leftarrow 0.5 * (r + (x/r))$

Retourner r

Exercice

En utilisant

L'affectation (\leftarrow),

Si... alors

Tant que... faire

Les opérateurs arithmétiques de base (+, -, *, /, et %)

Développez un algorithme en pseudo-code qui traduit un nombre écrit en nombre romain en nombre écrit en décimal.

Exercice

Le Si... alors n'est pas une opération essentielle.

En utilisant l'affectation et le Tant que... Faire, programmez le Si... Alors.

Plan de leçon

Qu'est-ce qu'un algorithme
Le pseudo-code
Exemples

Exactitude et efficacité (survol)

Les opérations de base

Les limites du pseudo-code

Un bon algorithme

Un bon algorithme est exact, efficace et élégant.

Exactitude et efficacité

Un algorithme est **exact** s'il donne la bonne sortie pour toute les entrées possibles. (Il ne se trompe jamais!)

Un algorithme est **efficace** s'il s'exécute en peu d'opérations basiques.

Un algorithme est **élégant** si ses instructions sont lisibles et peu nombreuses.

Exactitude et efficacité

Un algorithme est **exact** s'il donne la bonne sortie pour toute les entrées possibles. (Il ne se trompe jamais!)

Un algorithme est **efficace** s'il s'exécute en peu d'opérations basiques.

Comment calcule-ton l'efficacité?

Un algorithme est **élégant** si ses instructions sont lisibles et peu nombreuses.

Efficacité - Revenons au pgcd

```
i ← 1
pgcd ← 1
Tant que i ≤ x
Faire
    Si i divise x et y
    sans reste
        Alors pgcd ← i
    i ← i + 1
Retourner pgcd
```

**Combien d'opérations
pour**

x = 10, y = 8?

**Combien d'opérations
pour**

x = 3, y = 550?

y = 550, x = 3?

Le nombre d'opérations effectuées
dépend de l'entrée.

Efficacité - Revenons au pgcd

```
i ← 1
pgcd ← 1
Tant que i ≤ x
Faire
    Si i divise x et y
    sans reste
        Alors pgcd ← i
    i ← i + 1
Retourner pgcd
```

**Combien d'opérations
pour**

x = 10, y = 8?

**Combien d'opérations
pour**

x = 3, y = 550?

y = 550, x = 3?

Pouvez-vous faire mieux?

$O(2^n)$

Efficacité - Revenons au pgcd

$i \leftarrow x$

Si $(x > y)$, alors $i \leftarrow y$

Tant que $i > 1$, faire

 Si i divise x et y sans reste

 Alors retourner i

$i \leftarrow i - 1$

Retourner 1

Quel est le pire cas?
Le meilleur cas?

$O(2^n)$

Efficacité - Euclide

Tant que y est
différent de 0,

Faire

$t \leftarrow y$

$y \leftarrow x \bmod y$

$x \leftarrow t$

Retourner y

**Parcourir tous les
nombres entre 1 et x
est très coûteux.**

**Heureusement, Euclide
(Alexandrie, ≈ 300 av. J.-
C.) a trouvé mieux!**

$O(n^2)$

Plan de leçon

Qu'est-ce qu'un algorithme
Le pseudo-code
Exemples

Exactitude et efficacité (survol)

Les opérations de base

Les limites du pseudo-code

Qu'est-ce qu'une opération basique?

C'est une opération qui prend un temps constant.

Qu'est-ce qu'une opération basique?

C'est une opération qui prend un temps constant.

Est-ce que
« multiplier 698483 par 2929 »

prend le même temps que
« multiplier 5 par 6 »?

Est-ce que
« changer le signe de 68883838 »

prend le même temps que
« changer le signe de -7 »?

Qu'est-ce qu'une opération basique?

C'est une opération qui prend un temps constant.

Cela dépend du support de calcul

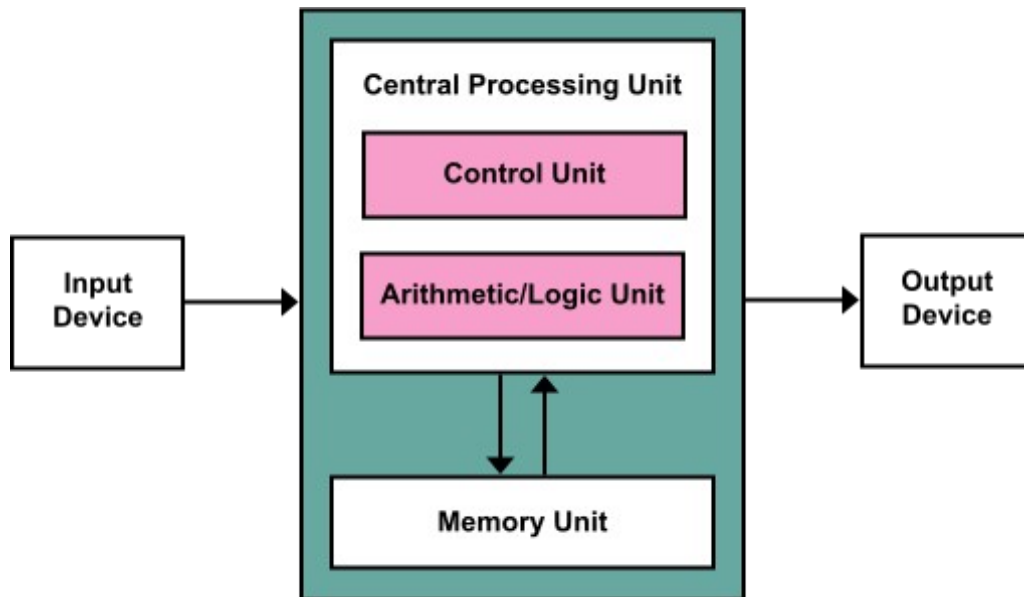


- Multiplier par 10.
- Diviser par 10.
- Comparer deux chiffres.
- Additionner deux chiffres.
- Lire un caractère.
- Changer le signe d'un nombre
- Etc...

Qu'est-ce qu'une opération basique?

C'est une opération qui prend un temps constant.

Cela dépend du support de calcul



- Multiplier par 2.
- Diviser par 2.
- Déplacer la tête de lecture.
- Changer le signe d'un registre.
- Additionner deux registres.
- Comparer deux registres.
- Etc...

Décomposition en opérations de base - La multiplication

La multiplication, en général, n'est pas une opération basique...

Entrée : Deux nombres naturels x et y

Sortie : Le produit des deux naturels.

$i \leftarrow 1$

$t \leftarrow x$

Tant que $i < y$

Faire

$x \leftarrow x + t$

$i \leftarrow i + 1$

Retourner x

Avec un papier et un crayon,
comment multipliez-vous?

Décomposition en opérations de base - La multiplication « à la russe »

Entrée : Deux nombres naturels x et y

Sortie : Le produit des deux naturels.

Les instructions:

$r = 0$

Tant que x est différent de 0, Faire

 Si x est impair, Alors

$r = r + y$

$x = x - 1$

$x = x / 2$

$y = y * 2$

Retourner r

En C, ça ressemble à quoi?

Plan de leçon

Qu'est-ce qu'un algorithme
Le pseudo-code
Exemples

Exactitude et efficacité (survol)

Les opérations de base

Les limites du pseudo-code

Le pseudo-code a ses limites...

Il est loin des processeurs et des registres :

Pas de débordement,

Pas d'erreur d'arrondi,

Pas de types,

La mémoire est arbitrairement grande,

Les opérations basiques ne sont pas basiques pour toutes les implémentations,

Chaque variable est accessible en la même quantité de temps,

...

Exercices

Traduisez les algorithmes du pseudo-code au C.

Algorithme d'Euclide, méthode de Babylone, multiplication à la russe et traduction de chiffres romains aux décimaux.

Testez-les.

**Avez-vous pensé à tous les cas limites?
(nombres positifs, négatifs, nuls, etc.)**

