



ALGORITHMIQUE

1. Définitions

Un algorithme est une description en langage naturel de la suite des actions effectuées par un programme. L'algorithme peut être :

- représenté graphiquement, c'est alors un algorigramme ou organigramme ;
- écrit sous une forme littérale, c'est le langage algorithmique.

1.1. L'algorigramme

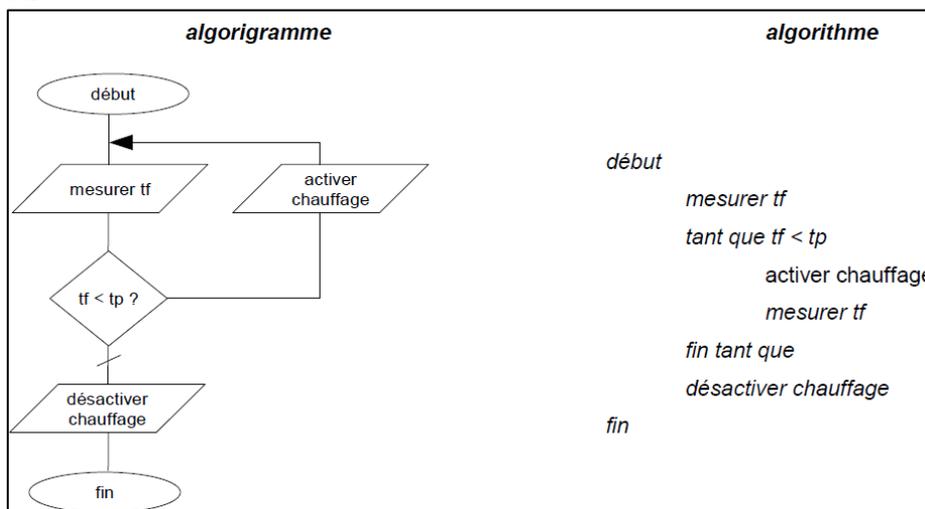
Les principaux symboles rencontrés dans un algorigramme sont représentés dans le tableau ci-dessous. L'algorigramme permet une vision globale mais reste limité aux études peu complexes.

SYMBOLE	DÉSIGNATION	SYMBOLE	DÉSIGNATION
	début ou fin d'un algorithme		Test ou Branchement conditionnel décision d'un choix parmi d'autres en fonction des conditions
	symbole général de « traitement » opération sur des données, instructions, ... ou opération pour laquelle il n'existe aucun symbole normalisé		sous-programme appel d'un sous-programme
	entrée / sortie		Liaison Les différents symboles sont reliés entre eux par des lignes de liaison. Le cheminement va de haut en bas et de gauche à droite. Un cheminement différent est indiqué à l'aide d'une flèche
	commentaire		

1.2. Le langage algorithmique

Le langage utilise un ensemble de mots clés (début, fin, faire, tant que, répéter, jusqu'à, ...). L'avantage de ce langage est sa transcription facile en langage de programmation dit évolué (Basic, Pascal, C, ...).

Exemple : un four à micro-ondes chauffe pendant un temps de fonctionnement t_f , jusqu'à ce que t_f atteigne le temps t_p , programmé par l'utilisateur.



1.3. Les différentes structures algorithmiques

Il existe trois structures algorithmiques différentes :

- la structure linéaire ou séquentielle ;
- les structures alternatives ou conditionnelles ;
- les structures répétitives ou itératives.

2. La structure linéaire

Les actions s'exécutent successivement dans l'ordre d'écriture.

Exemple d'une structure linéaire :

Un feu tricolore placé à un carrefour suit un même cycle à l'infini :

- feu vert allumé, feu orange éteint, feu rouge éteint ;
- temporisation de 20s
- feu vert éteint, feu orange allumé, feu rouge éteint ;
- temporisation de 5s
- feu vert éteint, feu orange éteint, feu rouge allumé ;
- temporisation de 20s

<u>algorithme</u>	<u>algorithme</u>

3. Les structures alternatives

Elle offre deux possibilités suivant une condition. Elle peut être de type complète ou réduite.

3.1. Structure alternative complète

L'exécution d'un des deux traitements dépend du résultat d'un test effectué sur une condition :

- si la condition est vraie, seule le premier traitement est exécuté;
- si la condition n'est pas vérifiée, seul est effectué le second traitement.

<u>algorithme</u>	<u>algorithme</u>
<pre> graph TD Start(()) --> Cond{condition ?} Cond --> Act1[action1] Cond --> Act2[action2] Act1 --> Join(()) Act2 --> Join Join --> End(()) </pre>	<pre> si condition alors action1 sinon action2 fin si </pre>

Exemple d'une structure alternative complète :

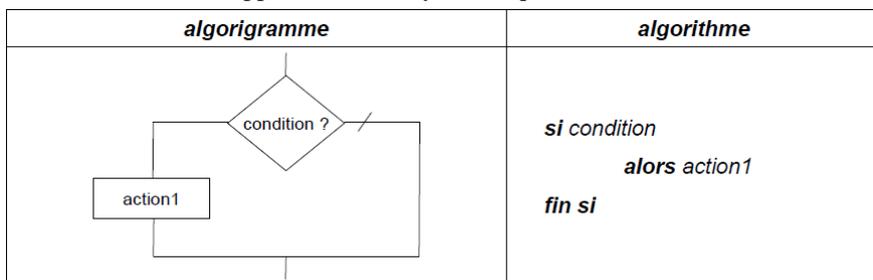
Tri de sacs : à la sortie de l'atelier de conditionnement d'une usine de fabrication d'engrais, un même convoyeur à bande transporte indifféremment des sacs de 25 Kg et 50 Kg. Un dispositif de tri automatique dirige ces sacs vers deux zones distinctes de stockage, D1 et D2.



<u>algorithme</u>	<u>algorithme</u>

3.2. Structure alternative réduite

Seule la condition vraie existe. La situation opposée conduit systématiquement à la sortie de la structure.



Exemple d'une structure alternative réduite :

Ouverture d'une porte de garage : le capteur de présence de la voiture du propriétaire du garage déclenche l'ouverture automatique de la porte.

<u>algorithme</u>	<u>algorithme</u>

4. Les structures répétitives

Il existe différents types de structures répétitives : soit le nombre de répétitions est connu, soit il est inconnu.

4.1. Le nombre de répétitions n'est pas connu

répéter ... jusqu'à ...		tant que ... faire ... fin tant que	
Le traitement est exécuté une première fois puis sa répétition se poursuit jusqu'à ce que la condition soit vérifiée.		On commence par tester la condition, si elle est vérifiée alors le traitement est exécuté tant que cette condition est vérifiée.	
<i>algorithme</i>	<i>algorithme</i>	<i>algorithme</i>	<i>algorithme</i>
	<p>répéter action jusqu'à condition vraie</p>		<p>tant que condition vraie faire action fin tant que</p>
L'action est toujours exécutée au moins une fois		L'action peut ne jamais être exécutée	

Exemple :

Alimentation en eau d'un lave-linge : une électrovanne assure le remplissage de la cuve d'un lave-linge jusqu'à ce qu'un capteur NH détecte le niveau haut.

<i>algorithme</i>	<i>algorithme</i>

Exemple :

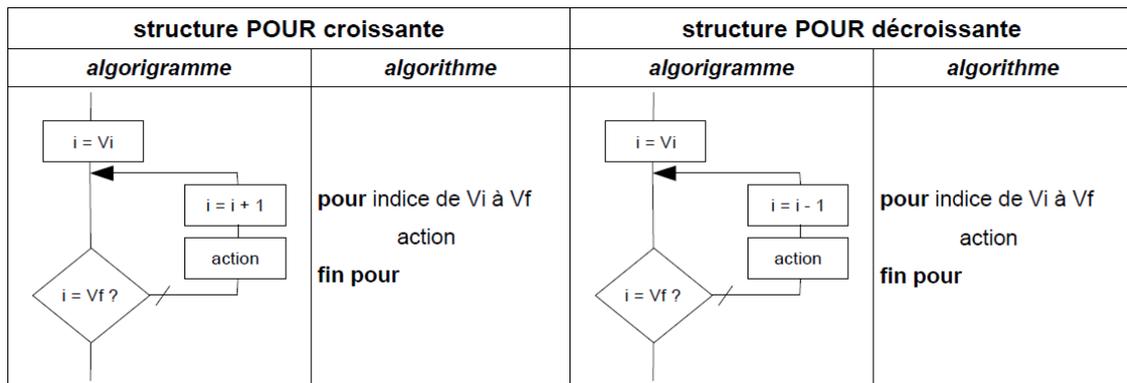
Chauffage d'un four : un four doit être chauffé à une température t de 500°C .

<i>algorithme</i>	<i>algorithme</i>

4.2. Le nombre de répétitions est connu

La sortie de la boucle d'itération s'effectue lorsque le nombre souhaité de répétitions est atteint. On utilise une variable (ou indice) de comptage d'itération, caractérisé par sa valeur initiale et sa valeur finale.

- Si la valeur finale est inférieure à la valeur initiale, la structure est dite décroissante.
- Si la valeur finale est supérieure à la valeur initiale, la structure est dite croissante.



Exemple :

Conditionnement de bouteilles : dans une usine de fabrication de jus de fruits les bouteilles sont conditionnées pas six, après contrôle, sous un film plastique rétractable.

<u>algorithme</u>	<u>algorithme</u>

TABLEAU DE SYNTHÈSE

	TYPE DE STRUCTURE ALGORITHMIQUE						
	LINÉAIRE	ITÉRATIVE = RÉPÉTITIVE				ALTERNATIVE	
		RÉPÉTER... JUSQU'À	TANT QUE FAIRE	RÉPÉTER... TANT QUE	POUR... RÉPÉTER	COMPLÈTE	RÉDUITE
MOTS CLÉS ET EXPRESSION		RÉPÉTER — TRAITEMENT JUSQU'À = CONDITION VRAIE -	TANT QUE = CONDITION VRAIE = FAIRE — TRAITEMENT	RÉPÉTER — TRAITEMENT TANT QUE = CONDITION VRAIE -	POUR V = V _i , A V = V _f RÉPÉTER — FAIRE V = V ± PAS — TRAITEMENT	SI - CONDITION VRAIE - ALORS — TRAITEMENT 1 SINON — TRAITEMENT 2	SI - CONDITION VRAIE - ALORS — TRAITEMENT
MOTS DÉLIMITEURS			FIN TANT QUE		FIN POUR	FIN SI	FIN SI
RÈGLES D'ÉCRITURE DE ALGORITHME	<ul style="list-style-type: none"> Les mots instructions sont écrits — au même rang — sur des lignes successives INSTRUCTION 1 INSTRUCTION 2 INSTRUCTION n	<ul style="list-style-type: none"> Les mots clés sont écrits : — au même rang — sur des lignes distinctes les mots instructions sont écrits : — entre les mots clés — sur des lignes successives RÉPÉTER INSTRUCTION 1 INSTRUCTION 2 INSTRUCTION n JUSQU'À = condition vraie =	<ul style="list-style-type: none"> Les mots clés sont écrits : — au même rang — sur des lignes distinctes les mots instructions sont écrits : — entre les mots clés — sur des lignes successives. le mot délimiteur FIN TANT QUE termine la structure TANT QUE = condition vraie = FAIRE INSTRUCTION 1 INSTRUCTION 2 INSTRUCTION n FIN TANT QUE	<ul style="list-style-type: none"> Les mots clés sont écrits : — au même rang — sur des lignes distinctes les mots instructions sont écrits : — entre les mots clés — sur des lignes successives. RÉPÉTER INSTRUCTION 1 INSTRUCTION 2 INSTRUCTION n TANT QUE = condition vraie =	<ul style="list-style-type: none"> Les mots clés sont écrits : — POUR et A sur une même ligne — RÉPÉTER sur la ligne suivante au même rang les mots instructions sont écrits : — entre les mots clés — sur des lignes successives — en commençant par l'expression de variation le mot délimiteur FIN POUR termine la structure POUR V = V_i, A V = V_f RÉPÉTER FAIRE V = V ± PAS TRAITEMENT FIN POUR 	<ul style="list-style-type: none"> les mots clés sont écrits : — SI sur une ligne — ALORS et SINON sur des lignes distinctes décalées d'un rang par rapport à SI les mots instructions sont écrits : — après ALORS — sur des lignes successives le mot délimiteur FIN SI termine la structure SI - condition vraie - ALORS 1 TRAITEMENT 1 SINON 1 TRAITEMENT 2 FIN SI 	<ul style="list-style-type: none"> les mots clés sont écrits : — SI sur une ligne — ALORS sur la ligne suivante décalée d'un rang par rapport à SI les mots instructions sont écrits : — après ALORS — sur des lignes successives le mot délimiteur FIN SI termine la structure SI - condition vraie - ALORS INSTRUCTION 1 INSTRUCTION 2 INSTRUCTION n FIN SI
ALGORIGRAMMES							