

ATELIER B

Didactique de l'algorithmique

Quelle pédagogie développer dans
le cadre de la présentation
d'algorithmes?

Martin QUINSON

Université de Lorraine

martin.quinson@loria.fr

Jean-Alain RODDIER

Rectorat de l'académie de Clermont-Ferrand

Jean-Alain.Roddier@ac-clermont.fr

Objectif de l'atelier

Recenser des méthodes pédagogiques développées (ou à développer) dans le cadre de la « présentation » d'algorithmes

Un outil pédagogique
pour l'enseignement
de l'algorithmique

6 phases :

- la découverte de l'algorithme ;
- l'appropriation de l'algorithme sur un exemple ;
- sa transcription en langage naturel ;
- son écriture en langage pseudocode ;
- son développement en langage de programmation ;
- l'observation de l'affectation des variables tout au long du déroulement du programme.

Un principe général

Toute mise en place

d'un point nouveau est exprimée
en parallèle d'un point d'appui.

1) La découverte de l'algorithme

Créer une situation d'appétence

2) L'appropriation de l'algorithme ^{1/2}

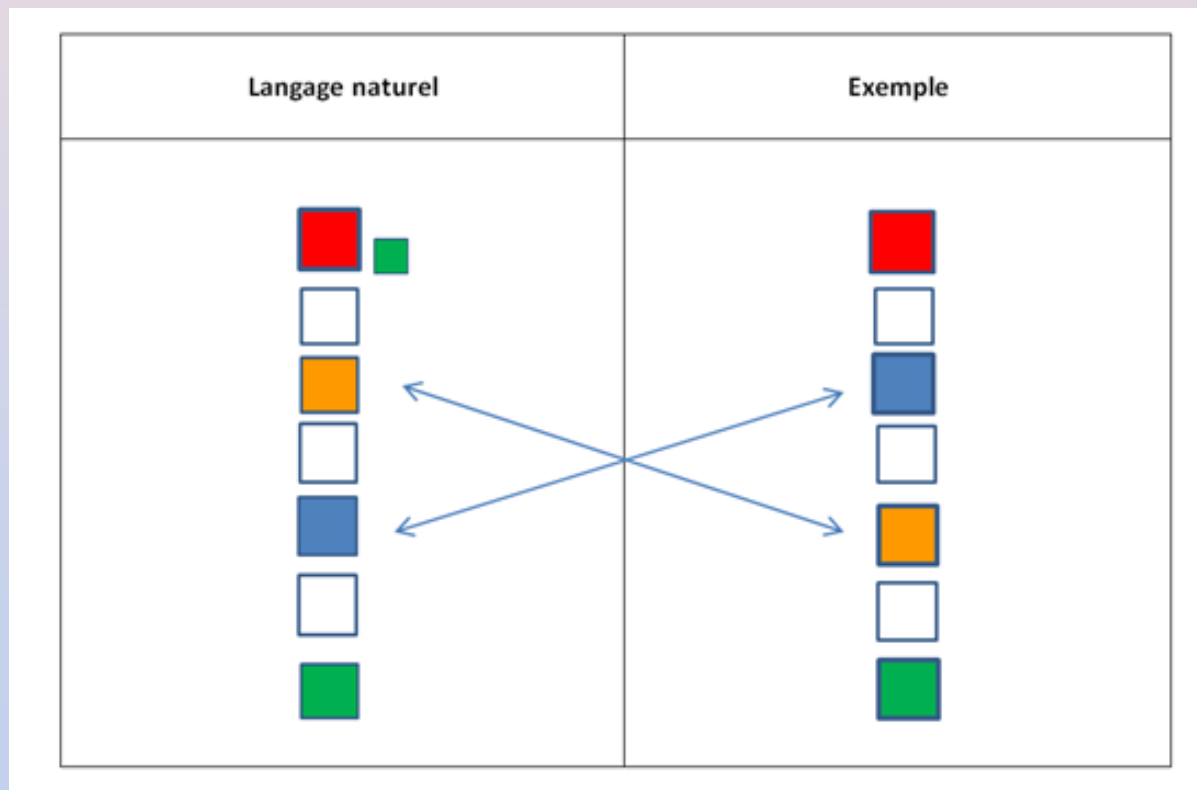
Mettre en avant le traitement d'un exemple

2) L'appropriation de l'algorithme ^{2/2}



- ❖ Traitement en continu et traitement pas à pas
- ❖ Des questions sur cet exemple
- ❖ La décortication « concrète » de l'exemple
 - Repasser en rouge l'élément de départ
 - Repasser en bleu le processus répétitif
 - Entourer en orange la condition d'arrêt
 - Repasser en vert l'élément produit à l'arrivée
- ❖ Le traitement par l'élève d'un exemple personnel

3) La traduction de l'algorithme en langage naturel



- ❑ Sa légitimité
- ❑ L'écriture en langage naturel



4) L'écriture de l'algorithme en langage pseudocode

Langage pseudocode	Langage naturel
	

5) Le développement en langage de programmation

Langage de programmation	Langage pseudocode
	

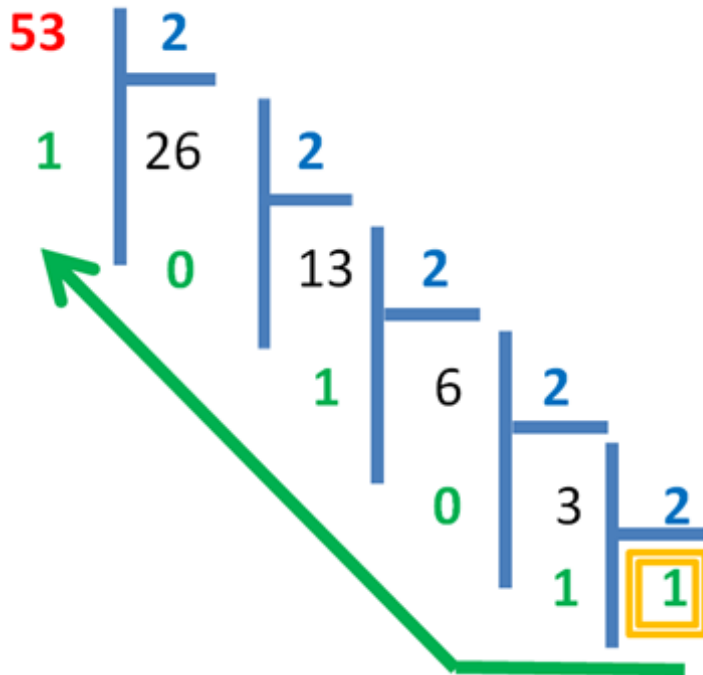
6) L'observation des valeurs des variables

Etape	N	i	k	s	L
0 (initialisation)		1	0	0	[]
1	(à remplir)	2	(à remplir)	(à remplir)	(à remplir)
2	"	"	"	"	"
(à remplir)	"	"	"	"	"
1000 (arrêt)	"	1001	"	"	

Une mise en œuvre de la méthode
sur un premier exemple :
le codage binaire

1) La découverte de l'algorithme

2) Le traitement d'un exemple : 53



53 divisé par 2 donne 26 reste 1

26 divisé par 2 donne 13 reste 0

13 divisé par 2 donne 6 reste 1

6 divisé par 2 donne 3 reste 0

3 divisé par 2 donne 1 reste 1

3) L'écriture en langage naturel

Langage naturel	Exemple
<p>On prend un entier N</p> <p>On le divise successivement par 2</p> <p>Jusqu'à obtenir un quotient égal à 1</p> <p>Le résultat est obtenu en mettant bout à bout 1 suivi des restes dans l'ordre inverse où on les a obtenus.</p>	<p>53</p> <p>53 divisé par 2 donne <u>26</u> reste 1</p> <p>26 divisé par 2 donne <u>13</u> reste 0</p> <p>13 divisé par 2 donne <u>6</u> reste 1</p> <p>6 divisé par 2 donne <u>3</u> reste 0</p> <p>3 divisé par 2 donne 1 reste 1</p> <p>1 1 0 1 0 1</p>

4) L'écriture en pseudocode

Pseudocode	Langage naturel
<p>Lire N</p> <p>Si $N=0$, renvoyer le résultat 0 Si $N=1$, renvoyer le résultat 1 $q \leftarrow E(N/2)$ et $r \leftarrow \text{Mod}(N,2)$ <u>listerésultat</u> $\leftarrow \{r\}$</p> <p>Répéter tant que $q \neq 1$ $N \leftarrow E(N/2)$ $q \leftarrow E(N/2)$ et $r \leftarrow \text{Mod}(N,2)$ <u>listerésultat</u> \leftarrow <u>listerésultat</u> + $\{r\}$ fin de tant que</p> <p><u>listerésultat</u> \leftarrow <u>listerésultat</u> + $\{1\}$ <u>listerésultat</u> \leftarrow <u>listerésultat</u> renversée Renvoyer la <u>listerésultat</u></p>	<p>On prend un entier N</p> <p>Si $N=0$, renvoyer le résultat 0 Si $N=1$, renvoyer le résultat 1 On commence par diviser N par 2 On prépare la liste résultat</p> <p>Jusqu'à obtenir un quotient égal à 1 On divise successivement par 2</p> <p>On prépare la liste résultat</p> <p>Le résultat est obtenu en mettant bout à bout 1 suivi des restes dans l'ordre inverse où on les a obtenus.</p>

5) Le développement en Python

Langage Python	Pseudocode
<pre>>>> def bin(N) : assert(N==int(N)) if N==0 : return 0 if N==1 : return 1 q = int(N/2) r = N%2 listerésultat = [r] while q != 1 : N = q q = int(N/2) r = N%2 listerésultat = listerésultat+[r] listerésultat = listerésultat+[1] listerésultat = listerésultat.reverse() return listerésultat</pre>	<p>On définit une fonction de N # On teste le caractère entier de N</p> <p>Si N=0, renvoyer le résultat 0 Si N=1, renvoyer le résultat 1 $q \leftarrow \underline{E(N/2)}$ $r \leftarrow \underline{\text{Mod}(N,2)}$ $\text{listerésultat} \leftarrow \{r\}$</p> <p>Répéter tant que $q \neq 1$</p> <p>$N \leftarrow \underline{E(N/2)}$ $q \leftarrow \underline{E(N/2)}$ $r \leftarrow \underline{\text{Mod}(N/2)}$ $\text{listerésultat} \leftarrow \text{listerésultat} + \{r\}$</p> <p>fin de tant que $\text{listerésultat} \leftarrow \text{listerésultat} + \{1\}$ $\text{listerésultat} \leftarrow \text{listerésultat renversée}$ Renvoyer la liste listerésultat</p>

6) La gestion des variables

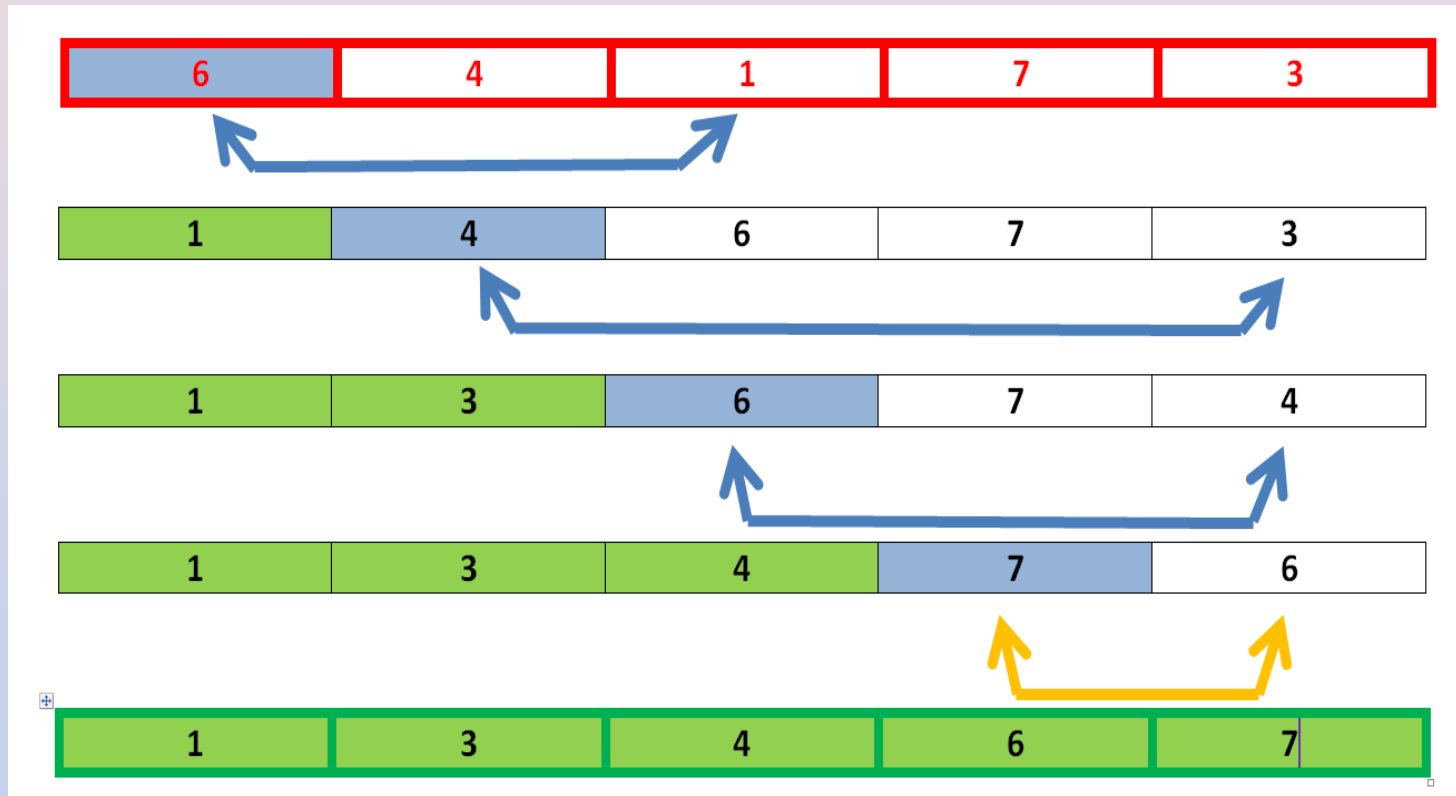
N° de l'étape	N	q	r	listeresultat
0	53	26	1	[1]
1	26	13	0	[0 , 1]
2	13	6	1	[1 , 0 ,1]
3	6	3	0	[0 , 1 , 0 , 1]
4	3	1	1	[1, 0 , 1 , 0 , 1]
5				[1,1,0,1,0,1]
6				[1,0,1,0,1,1]

**Une mise en œuvre de la méthode
sur un deuxième exemple :
le tri par sélection**

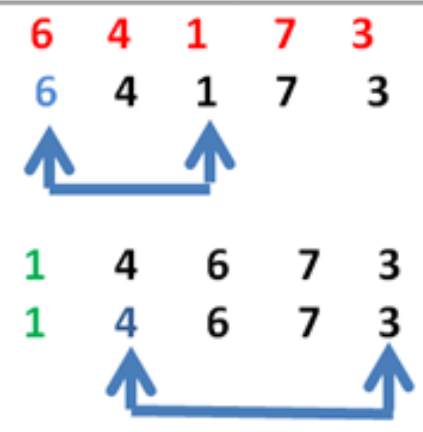

1) La découverte de l'algorithme

2) Le traitement d'un exemple

6	4	1	7	3
---	---	---	---	---



3) L'écriture de l'algorithme en langage naturel

Langage naturel	Exemple
<p>On part d'une liste</p> <p>On sélectionne le premier élément de la liste et on l'inverse (si besoin est) avec l'élément le plus petit du reste de la liste.</p> <p>On sélectionne le deuxième élément de la liste et on l'inverse (si besoin est) avec l'élément le plus petit du reste de la liste.</p>	 <p>6 4 1 7 3 6 4 1 7 3</p> <p>1 4 6 7 3 1 4 6 7 3</p>
<p>On poursuit le même procédé</p> <p>jusqu'à sélectionner l'avant dernier élément de la liste</p> <p>et l'inverser si besoin est avec le dernier élément de la liste.</p> <p>On obtient la liste triée dans l'ordre croissant.</p>	 <p>1 3 6 7 4 1 3 4 7 6</p> <p>1 3 4 7 6 1 3 4 7 6</p> <p>1 3 4 6 7</p>

4) L'écriture en pseudocode

Pseudocode	Langage naturel
<p>Lire la liste T</p> <p>$n \leftarrow$ longueur de la liste T</p> <p>Pour i variant de 0 jusqu'à n-2</p> <p> $imin \leftarrow i$</p> <p> $Tmin \leftarrow T[i]$</p> <p> Pour j variant de i+1 à n-1</p> <p> Si $T[j] > Tmin$ alors</p> <p> $imin \leftarrow j$</p> <p> $Tmin \leftarrow T[j]$</p> <p> Prendre le j suivant</p> <p> $T[i], T[imin] \leftarrow Tmin, T[i]$</p> <p>Prendre le i suivant</p> <p>Écrire la liste T</p>	<p>On part d'une liste (on la note T)</p> <p>On note n la longueur de la liste</p> <p>On sélectionne le premier élément (<i>c'est T[0]</i>) de la liste et on l'inverse (si besoin est) avec l'élément le plus petit (<i>Tmin</i>) du reste de la liste.</p> <p>On sélectionne le deuxième élément de la liste (<i>c'est T[1]</i>) et on l'inverse (si besoin est) avec l'élément le plus petit (<i>Tmin</i>) du reste de la liste.</p> <p>On poursuit le même procédé jusqu'à sélectionner l'avant dernier élément de la liste (<i>c'est T[n-2]</i>) et l'inverser si besoin est avec le dernier élément de la liste.</p> <p>On obtient la liste triée dans l'ordre croissant.</p>

5) Le développement en langage Python

Langage Python	Pseudocode
<pre>>>> def tri(T) : n=len(T) for i in range (0, n-1) : # sur python, i varie de 0 à n-2 (compris) imin = i Tmin = T[i] For j in range(i+1, n) : # j varie de i+1 à n -1 If T[j]<Tmin : imin =j Tmin = T[j] T[i],T[imin] = Tmin,T[i] return(T)</pre>	<p>Lire la liste T (entrer cette liste sous la forme d'un n-uplet) $n \leftarrow$ longueur de la liste T</p> <p>Pour i variant de 0 jusqu'à n-2</p> <p>$imin \leftarrow i$ $Tmin \leftarrow T[i]$ Pour j variant de i+1 à n-1</p> <p>Si $T[j] < Tmin$ alors $imin \leftarrow j$ $Tmin \leftarrow T[j]$</p> <p>Prendre le j suivant $T[i], T[imin] \leftarrow Tmin, T[i]$ Prendre le i suivant Renvoyer la liste T</p>

6) La gestion des variables

N° étape	T	i	<u>imin</u>	<u>Tmin</u>	j	T[j]
0	6 4 1 7 3	0	0	6	1	4
1	6 4 1 7 3	0	1	4	2	1
2	6 4 1 7 3	0	2	1	3	7
3	6 4 1 7 3	0	2	1	4	3
4	1 4 6 7 3	1	1	4	2	6
5	1 4 6 7 3	1	1	4	3	7
6	1 4 6 7 3	1	1	4	4	3
7	1 3 6 7 4	2	2	6	3	7
8	1 3 6 7 4	2	2	6	4	4
9	1 3 4 7 6	3	3	7	4	6
10	1 3 4 6 7	(4)				

Conclusion

Inscrire la présentation des algorithmes
dans une véritable dynamique explicative
où les implicites sont levés.

Le caractère évolutif de ce texte.

Fin de cette première présentation