

Jean-Michel Léry

2<sup>e</sup>  
édition

# Algorithmique en C, C++, Java, Python et PHP



# Table des matières

<b>Introduction .....</b>	<b>1</b>
<b>Environnement et conventions .....</b>	<b>7</b>
<b>1. Les étapes de développement d'une application .....</b>	<b>7</b>
1.1. L'analyse.....	7
1.2. L'algorithme .....	10
1.3. L'écriture du programme dans un langage de programmation .....	11
<b>2. Le pseudo-langage.....</b>	<b>18</b>
2.1. Son rôle .....	18
2.2. Conventions syntaxiques.....	18
<b>3. Le génie logiciel .....</b>	<b>20</b>
3.1. La modélisation des données .....	20
3.2. La modularité du programme .....	20
3.3. La lisibilité du programme.....	20
3.4. Privilégier les solutions simples.....	21
3.5. L'utilisation de conventions d'écriture.....	21
<b>4. La performance algorithmique .....</b>	<b>21</b>
<b>Résumé .....</b>	<b>22</b>
<b>Exercices .....</b>	<b>22</b>
<b>Solutions.....</b>	<b>23</b>
<b>Les traitements logiques .....</b>	<b>29</b>
<b>1. Les tests ou instructions conditionnelles .....</b>	<b>29</b>
1.1. Présentation .....	29
1.2. La logique booléenne .....	30
1.3. L'instruction SI ALORS SINON .....	31
1.4. Le choix multiple .....	33
<b>2. Les boucles ou instructions répétitives .....</b>	<b>37</b>
2.1. Présentation des boucles .....	37
2.2. Comprendre le fonctionnement d'une boucle.....	46
2.3. Construire une boucle .....	49
<b>3. Les sous-programmes.....</b>	<b>54</b>
3.1. Les procédures .....	54
3.2. Les fonctions .....	54
3.3. Organisation en modules autonomes .....	55
<b>4. La complexité algorithmique .....</b>	<b>61</b>
4.1. Principe .....	61
4.2. L'analyse de la complexité .....	62
4.3. Les grandes familles de complexité.....	62
4.4. Choisir un algorithme .....	64
<b>Résumé .....</b>	<b>64</b>

Exercices .....	64
Solutions .....	67
<b>La gestion des données .....</b>	<b>93</b>
<b>1. Principe de traitement des données .....</b>	<b>93</b>
1.1. Le rôle des variables .....	93
1.2. Le rôle des fichiers .....	94
<b>2. Les tableaux .....</b>	<b>94</b>
2.1. Description .....	94
2.2. L'allocation mémoire .....	95
2.3. Gestion d'un tableau .....	96
<b>3. Les enregistrements .....</b>	<b>121</b>
3.1. Description .....	122
3.2. Gestion d'un enregistrement .....	122
<b>4. Les pointeurs .....</b>	<b>126</b>
4.1. Description .....	126
4.2. Les pointeurs et l'allocation statique de la mémoire .....	127
4.3. L'allocation dynamique de la mémoire .....	129
4.4. La particularité de Java .....	130
4.5. La particularité de PHP .....	130
4.6. La particularité de Python .....	130
<b>5. Les listes chaînées .....</b>	<b>130</b>
5.1. Description .....	130
5.2. L'allocation mémoire .....	131
5.3. Gestion d'une liste chaînée .....	132
<b>6. Gestion des données complexes .....</b>	<b>159</b>
6.1. Les tableaux d'enregistrements .....	159
6.2. Les listes d'enregistrements .....	165
<b>7. Variantes sur les tableaux .....</b>	<b>166</b>
7.1. Les tableaux dynamiques .....	166
7.2. Les tableaux de pointeurs .....	170
<b>Résumé .....</b>	<b>172</b>
<b>Exercices .....</b>	<b>172</b>
<b>Solutions .....</b>	<b>173</b>
<b>La récursivité .....</b>	<b>195</b>
<b>1. Principe .....</b>	<b>195</b>
<b>2. Diviser pour résoudre .....</b>	<b>200</b>
<b>3. Suppression de la récursion .....</b>	<b>201</b>
<b>4. Récursivité croisée .....</b>	<b>203</b>
<b>Résumé .....</b>	<b>206</b>
<b>Exercices .....</b>	<b>206</b>
<b>Solutions .....</b>	<b>208</b>
<b>Les données abstraites .....</b>	<b>219</b>
<b>1. Les piles .....</b>	<b>219</b>
1.1. Principe .....	219
1.2. Les primitives .....	220

1.3. Utilisation des tableaux.....	220
1.4. Utilisation des listes chaînées .....	221
1.5. Exemples .....	223
1.6. Les piles et la récursivité .....	234
1.7. La classe <i>C++ stack</i> .....	236
1.8. La classe <i>Java Stack</i> .....	236
1.9. La classe <i>PHP SplStack</i> .....	237
1.10. Les listes <i>Python en Piles</i> .....	238
1.11. La classe <i>Python deque</i> .....	238
<b>2. Les files .....</b>	<b>238</b>
2.1. Principe .....	238
2.2. Utilisation des tableaux.....	239
2.3. Utilisation des listes chaînées .....	240
2.4. La classe <i>C++ deque</i> .....	241
2.5. L'interface <i>Java Queue</i> avec la classe <i>LinkedList</i> .....	242
2.6. La classe <i>PHP SplQueue</i> .....	243
2.7. Les listes <i>Python en Files</i> .....	244
2.8. La classe <i>Python deque</i> .....	244
<b>3. Les arbres .....</b>	<b>245</b>
3.1. Principe et définitions .....	245
3.2. Les types d'arbres.....	246
3.3. Création d'un arbre binaire d'expression.....	247
3.4. Parcours d'un arbre binaire.....	249
3.5. Les arbres de recherche .....	268
<b>Résumé .....</b>	<b>269</b>
<b>Exercices .....</b>	<b>270</b>
<b>Solutions .....</b>	<b>271</b>
<b>Les tris .....</b>	<b>317</b>
<b>1. Tris élémentaires.....</b>	<b>317</b>
1.1. Le tri par sélection.....	317
1.2. Le tri par insertion .....	325
1.3. Le tri à bulles .....	329
1.4. Le tri par comptage .....	333
<b>2. Tris avancés.....</b>	<b>334</b>
2.1. Le tri shell .....	334
2.2. Le tri indirect .....	340
2.3. Le tri rapide ou <i>quicksort</i> .....	351
<b>Résumé .....</b>	<b>359</b>
<b>Exercices .....</b>	<b>359</b>
<b>Solutions .....</b>	<b>360</b>
<b>Les recherches.....</b>	<b>371</b>
<b>1. La recherche séquentielle.....</b>	<b>371</b>
<b>2. La recherche dichotomique .....</b>	<b>376</b>
2.1. Principe .....	376
2.2. Version récursive.....	376
2.3. Version itérative .....	381
<b>3. La recherche par interpolation .....</b>	<b>382</b>

<b>4. Tables de hachage ou adressage dispersé.....</b>	<b>383</b>
4.1. Principe .....	383
4.2. La fonction de hachage .....	384
4.3. Table de hachage chaînée.....	386
4.4. Table de hachage à adressage ouvert .....	403
<b>5. Les arbres de recherche équilibrés.....</b>	<b>407</b>
5.1. Principe .....	407
5.2. Fonctionnement d'un arbre AVL .....	408
5.3. Gestion du facteur d'équilibre .....	409
5.4. Rotations des arbres AVL .....	410
<b>Résumé .....</b>	<b>414</b>
<b>Exercices .....</b>	<b>415</b>
<b>Solutions.....</b>	<b>416</b>
<b>Les méthodes numériques .....</b>	<b>463</b>
<b>1. Interpolation polynomiale.....</b>	<b>463</b>
1.1. Principe .....	463
1.2. Rappel sur les polynômes.....	464
1.3. Le polynôme d'interpolation.....	465
1.4. Le programme.....	469
1.5. La fonction <i>interpolate.interp1d</i> de la bibliothèque <i>Python scipy</i> .....	484
<b>2. Méthode des moindres carrés .....</b>	<b>486</b>
2.1. Principe .....	486
2.2. Rappel sur le calcul des distances .....	487
2.3. Le programme.....	488
2.4. La fonction <i>linalg.lstsq</i> de la bibliothèque <i>Python numpy</i> .....	498
<b>3. Recherche des solutions d'équations.....</b>	<b>500</b>
3.1. Principe de la méthode de Newton.....	500
3.2. Rappel sur les dérivées.....	501
3.3. Choix du point de départ pour la méthode de Newton .....	504
3.4. Le programme.....	506
3.5. La fonction <i>optimize.bisect</i> de la bibliothèque <i>Python scipy</i> .....	541
<b>Résumé .....</b>	<b>542</b>
<b>Les algorithmes classiques.....</b>	<b>543</b>
<b>1. Algorithme du plus court chemin de Dijkstra .....</b>	<b>543</b>
1.1. Rappel sur les graphes .....	543
1.2. Description de l'algorithme.....	544
1.3. L'algorithme .....	547
1.4. Application au calcul d'itinéraire.....	573
<b>2. Algorithme de compression de données de Huffman.....</b>	<b>574</b>
2.1. Principe de la compression de données.....	574
2.2. Description de l'algorithme.....	575
2.3. L'algorithme .....	579
<b>Résumé .....</b>	<b>637</b>
<b>Index.....</b>	<b>639</b>