



Fenofitia Nomenjanahary

Kajy University: Informatique

Algorithmes et structures de données

Author: Dimby Rabearivony

Date: April 16, 2024

Version: 1.0



"Ny fahalalana no valamparihiko."

Contents

1	Structures de données fondamentales	2
1.1	Variables et langage de programmation	2
1.2	Pointeurs	2
1.3	Types de données	2
1.4	Tableaux	2
1.5	Chaînes de caractères	2
1.6	Struct	2
1.7	Exercices	2
2	Algorithmes de recherche et de tri	3
2.1	Qu'est-ce qu'un algorithme?	3
2.2	Tri	3
2.3	Algorithmes de recherche	3
2.4	Exercices	3
3	Complexité algorithmique	4
3.1	Récursion	4
3.2	La notation O	4
3.3	Les notations Ω et θ	4
3.4	Complexité de certains algorithmes	4
3.5	Exercices	4
4	Structures de données avancées	5
4.1	Listes chaînées	5
4.2	Piles	5
4.3	Files	5
4.4	Arbres binaires	5
4.5	Arbres	5
4.6	Graphes	5
4.7	Exercices	5
5	Applications d'algorithmes en IA	6
5.1	Greedy algorithm	6

Introduction

“En fait, je dirai que la différence entre un mauvais programmeur et un bon réside dans le fait qu’il considère son code ou ses structures de données comme plus importants. Les mauvais programmeurs se soucient du code. Les bons programmeurs se soucient des structures de données et de leurs relations.”

- Linus Torvalds

Chapter 1 Structures de données fondamentales

1.1 Variables et langage de programmation

Avant de passer à la définition des variables, mettons-les en relation avec d'anciennes équations mathématiques. Nous avons tous résolu de nombreuses équations mathématiques depuis notre enfance. À titre d'exemple, considérons l'équation ci-dessous :

$$3x + 4y = 5$$

Nous n'avons pas à nous soucier de l'utilisation de cette équation. La chose importante que nous devons comprendre est que l'équation a des noms (x et y) qui contiennent des valeurs (données). Cela signifie que les noms (x et y) sont des espaces réservés pour représenter les données. De même, en programmation informatique, nous avons besoin de quelque chose pour conserver les données, et les variables sont le moyen d'y parvenir.

1.2 Pointeurs

1.3 Types de données

1.4 Tableaux

1.5 Chaînes de caractères

1.6 Struct

1.7 Exercices

Chapter 2 Algorithmes de recherche et de tri

2.1 Qu'est-ce qu'un algorithme?

Définition 2.1. Algorithme

Un algorithme est constitué d'instructions étape par étape sans ambiguïté pour résoudre un problème donné.



Exemple 2.1 Considérons le problème de la préparation d'une omelette. Pour préparer une omelette, on suit la étapes indiquées ci-dessous :

1. Récupérez la poêle.
2. Récupérez l'huile.
 - (a). Avons-nous de l'huile?
 - I. Si oui, mettez-le dans la poêle.
 - II. Si non, voulons-nous acheter de l'huile ?
 - A. Si oui, sortez et achetez.
 - B. Si non, on termine.
3. Allumez la cuisinière, etc...

Ce que nous faisons, c'est que, pour un problème donné (préparer une omelette), nous proposons une procédure étape par étape pour le résoudre.

2.2 Tri

2.2.1 Tri à bulle

2.2.2 Tri par tas

2.3 Algorithmes de recherche

2.3.1 Recherche systématique

2.3.2 Recherche par

2.4 Exercices

Chapter 3 Complexité algorithmique

3.1 Récursion

3.2 La notation O

3.3 Les notations Ω et θ

3.4 Complexité de certains algorithmes

3.5 Exercices

Chapter 4 Structures de données avancées

4.1 Listes chaînées

4.2 Piles

4.3 Files

4.4 Arbres binaires

4.5 Arbres

4.6 Graphes

4.7 Exercices

Chapter 5 Applications d'algorithmes en IA

5.1 Greedy algorithm

Bibliography

- [1] Karumanchi, N. (2017). Data structures and algorithms made easy.
- [2] Knuth, D. E. (1986). *The T_EX Book*. Addison-Wesley Professional.