

# Étude et Comparaison d'Algorithmes

Thibault BEZIER LA FOSSE

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Objectif . . . . .	3
1.2	Outils utilisés . . . . .	3
1.3	Structure du programme . . . . .	3
<b>2</b>	<b>Algorithme 1 : PairAlgorithm</b>	<b>4</b>
2.1	Implémentation . . . . .	4

# 1 Introduction

Ce projet en six séances a pour objectif d'implémenter des algorithmes, afin de calculer leurs complexités et comparer les performances. Quatre algorithmes sont à notre disposition, ce compte rendu les traitera dans l'ordre donné. Seront expliqués :

L'implémentation

La complexité théorique

Les résultats d'exécutions

L'analyse des résultats

théoriques et expérimentaux

## 1.1 Objectif

Les quatre algorithmes à implémenter consistent à trouver la suite de termes du tableau dont la somme est maximum. Les éléments du tableau sont choisis de manière aléatoire, et nous exécuterons nos algorithmes sur des tableaux de tailles variables, en commençant par des tableaux de moins d'une dizaine d'éléments, à des tableaux qui en contiennent des dizaines de milliers.

## 1.2 Outils utilisés

Le langage de programmation à utiliser pour ce projet étant libre, j'ai choisi de m'orienter vers le C++. D'abord pour une raison pratique, l'université nous fait régulièrement utiliser ce langage. Ensuite pour des raisons de performances, car C++ reste un langage assez efficace pour effectuer des opérations comme celles de ce projet. Le compilateur utilisé est G++, avec la bibliothèque C++11. Effectivement, G++ est disponible à l'université, et la bibliothèque C++11 offre des méthodes efficaces, et surtout pratiques pour manipuler des tableaux.

## 1.3 Structure du programme

Quatre algorithmes étant à effectuer, j'ai choisi de créer une arborescence de classes sur un seul niveau. En haut, une classe abstraite implémentant les principales méthodes nécessaires à l'exécution des algorithmes, et en bas les algorithmes qui redéfinissent la méthode de calcul de la méthode MAXSOMME.

## 2 Algorithme 1 : PairAlgorithm

### 2.1 Implémentation