

Réalisation d'un programme pour apprendre les algorithmes de tris

PROJET DE DEVELOPPEMENT LOGICIEL

MEHDAOUI Abderrahim, ZEISEL Helmut Michael
CENTRALESUPELEC

Table des matières

Introduction.....	2
Recherche d'informations et sélection des outils disponibles	3
Structure du code	3
Conclusion	5

Introduction

Un algorithme de tri permet d'organiser une collection d'objets selon un critère donné. Ils sont utilisés dans de très nombreuses situations et particulièrement utiles à de nombreux algorithmes plus complexes dont les algorithmes de recherches comme l'algorithme de recherche par dichotomie. Ils peuvent également servir à mieux organiser les données pour faciliter leur utilisation futures.

La collection à trier est souvent donnée sous forme de tableau/liste, afin de permettre un accès simplifié aux différents éléments de la collection.

Bon nombre d'algorithmes de tri procèdent par comparaisons successives, et peuvent donc être définis indépendamment de l'ensemble auquel appartiennent les éléments et de la relation d'ordre associée. Un même algorithme peut par exemple être utilisé pour trier des réels et des chaînes de caractères.

Les algorithmes de tris sont souvent étudiés dans les cours d'informatique pour introduire des notions comme la complexité.

Dans ce projet nous nous intéresserons à quelques algorithmes connues dont ceux de « tri rapide, » « tri insertion » en fonction du temps restants. En accord avec les contraintes du sujet les données en entrées seront exclusivement des entiers naturels.

Nous détaillerons dans un premier temps notre recherche d'informations ainsi que les outils sélectionnés puis la structure générale du code. Nous conclurons enfin ce rapport par une opinion général et un peu de recul sur le projet.

Recherche d'informations et sélection des outils disponibles

Cette étape du projet consiste à recueillir des informations sur les algorithmes de tri, les langages pouvant être utilisés ainsi que leurs environnements de développements, pour avoir un premier aperçu de ces derniers et sélectionner les outils qui seront utilisés durant le développement. Cette étape s'est déroulée sans encombre, la majorité des informations à connaître sur les algorithmes a été recueillie via des recherches sur internet afin de retrouver les acquis d'il y a quelques années. Il nous a semblé utile pour notre apprentissage d'utiliser le langage python, chaque membre du binôme l'ayant utilisés durant les différents travaux de laboratoire au cours de son cursus respectif, ce projet était l'occasion de faire de la programmation objets en python pour l'un des membres du binôme. Nous avons par ailleurs choisi d'utiliser l'environnement de développement PyCharm pour réaliser ce projet, il semble complet et reconnu. En ce qui concerne les bibliothèques utilisées nous sommes restés dans ce qu'il y a de connu comme tkinter (ou ttk pour avoir un rendu graphique plus moderne), panda pour gérer la saisie de données sur Excel et d'autres bibliothèques comme numpy.

Structure du code

Nous avons choisi de structurer notre code en tentant de nous rapprocher de l'architecture modèle-vue-contrôleur.

Le modèle rassemble les classes regroupant les algorithmes, la vue regroupe les classes gérant l'affichage et le contrôleur est composé des classes main et config.

Ci-dessous un diagramme de classes permettant de mieux comprendre la structure du code et les composantes des classes.

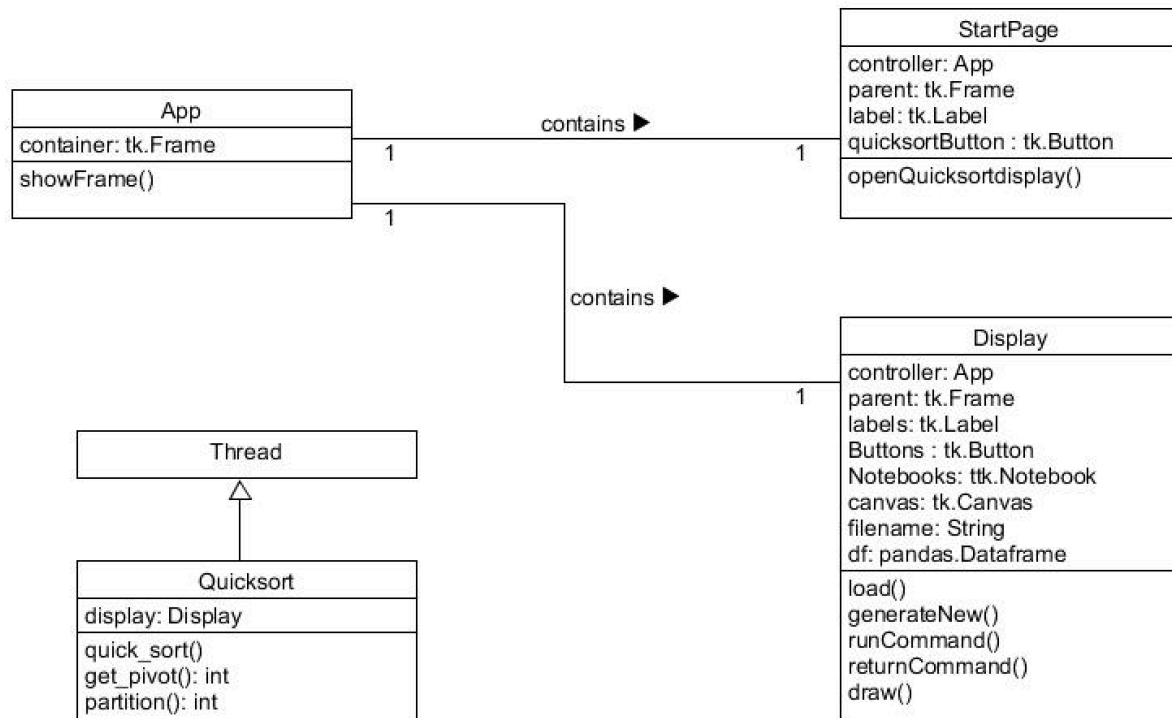


Diagramme de classe représentant l'architecture général du programme

Nous avons rencontré quelques problèmes sur la manière de réaliser l'animation mais après quelques recherches sur internet, il nous a semblé judicieux d'utiliser la bibliothèque tkinter qui possède de nombreuses fonctionnalités liées aux affichages d'animation. L'un des membres du binôme connaissait déjà cette bibliothèque ce qui a ensuite facilité l'implémentation de l'animation.

En ce qui concerne le visuel général de l'application, nous souhaitons afficher un premier menu qui permet à l'utilisateur de choisir quel algorithme il souhaite étudier. Une fois l'algorithme sélectionné une nouvelle fenêtre affiche une description de l'algorithme, son code en pseudo code ainsi qu'une animation permettant de comprendre le fonctionnement de l'algorithme via un exemple. L'utilisateur peut choisir de lancer l'animation avec une liste aléatoire ou de charger sa propre liste à partir d'un fichier Excel. Pour traiter les données rentrées par l'utilisateur nous avons choisi d'utiliser la bibliothèque Pandas qui permet de traiter de larges volumes de données, elle semble donc légèrement trop sophistiquée pour le projet mais les fonctions disponibles nous permettent de produire un programme répondant aux normes du sujet. Pour réaliser l'animation nous avons choisi d'afficher un histogramme de valeurs aléatoires comprises entre 0 et 100. Une fois l'application lancée, chaque rectangle de l'histogramme bouge dynamiquement en fonction du déroulement de l'algorithme choisi. A la demande de l'enseignant et pour rendre l'animation plus claire nous avons affiché sous chaque rectangle le nombre qui lui est associé. Nous avons également trouvé une méthode pour rentrer un texte descriptif bien présenté en utilisant le langage HTML afin d'améliorer son esthétique.

A la fin du projet nous avons rencontré quelques problèmes d'affichage car certains fonts n'étaient pas compatibles entre Linux et Mac OS. Les fonts permettant de choisir la police de caractères du texte des descriptions ou des boutons. Pour gérer ce problème nous avons choisi de coder la suite du projet sur une seule machine.

Conclusion

Les contraintes temporelles ne nous ont permis d'implémenter certaines fonctionnalités. Tout d'abord nous n'avons pu décrire que trois algorithmes, néanmoins le modèle utilisé ne change pas et il est alors simple d'ajouter autant d'algorithmes que nécessaire. Nous souhaitons également pouvoir lancer l'application depuis un icône comme sur Windows mais cela s'implémente différemment sur mac, nous avons donc abandonné cette idée par manque de temps également. Enfin, nous n'avons pas encore implémenter la comparaison des différents algorithmes décrit à l'heure ou ce rapport est rédigé. Nous espérons cependant implémenter cette fonctionnalité avant la soutenance finale afin d'améliorer notre programme.

Avec un peu de recul nous avons pris du plaisir à réaliser ce programme, nous avons au départ des attentes élever en terme de design et de fonctionnalités mais malheureusement le temps ne nous a pas permis d'atteindre ces objectifs. Le produit final répond tout de même aux normes du sujet.