Rapport P21 v2 - FOURMIS

1. Choix / Fourmis Travailleuse

1.1 Algorithme de trie

La fourmis travailleuse se déplace une à une lorsqu'elle cherche de la nourriture, mais ce déplacement est dicté par un "hasard" : il doit privilégier les nœuds contenant le plus de phéromones. Alors pour appliquer l'algorithme mentionné dans le sujet nous avons dû utiliser un Collections.shuffle (mélange notre liste) et un Collections.sort (trie notre liste), mais pour que le trie marche correctement (c'est à dire que la liste soit trié par ordre croissant de phéromone) nous avons dû implémenter l'interface Comparator et override la fonction compareTo.
Car le collections.sort() appelle implicitement compareTo. Alors il faut le remanier pour qu'il prenne en paramètre les quantités de phéromones, et cela s'avère bien pratique !

1.2 Rencontre d'un objet lors du chemin retour

Lorsque la fourmis travaille à trouver de la nourriture, son mouvement est centré sur la liste historique. Mais lorsque la fourmis rencontre un obstacle sur son chemin, nous avons choisi de faire en sorte que la fourmis se libère de son chemin historique et ce déplacement aléatoirement sur les nœuds jusqu'à retrouver sa fourmilière.

Nos difficultés :

- De voir si la fonction move() de worker fonctionnait (une fonction assez compliquée), car avec la fatigue nous oublions des choses essentielles et assez faciles et la longueur du code, en effet en fonction du temps nous avons vite oublié des détails important.

- Lorsque nous étions à notre 4ème ou 5ème heure de travail, il était assez difficile de prendre du recul, et nous oublions certaines choses comme remplir les bitsets, ce qui nous bloquait un certain temps car nous comprenions pas pourquoi les tests (qui utilisait les bitsets) ne fonctionnait pas.

- Par rapport à la création du fichier, la partie délicate était de faire en sorte que les données récupérées soit cohérente avec les itérations.