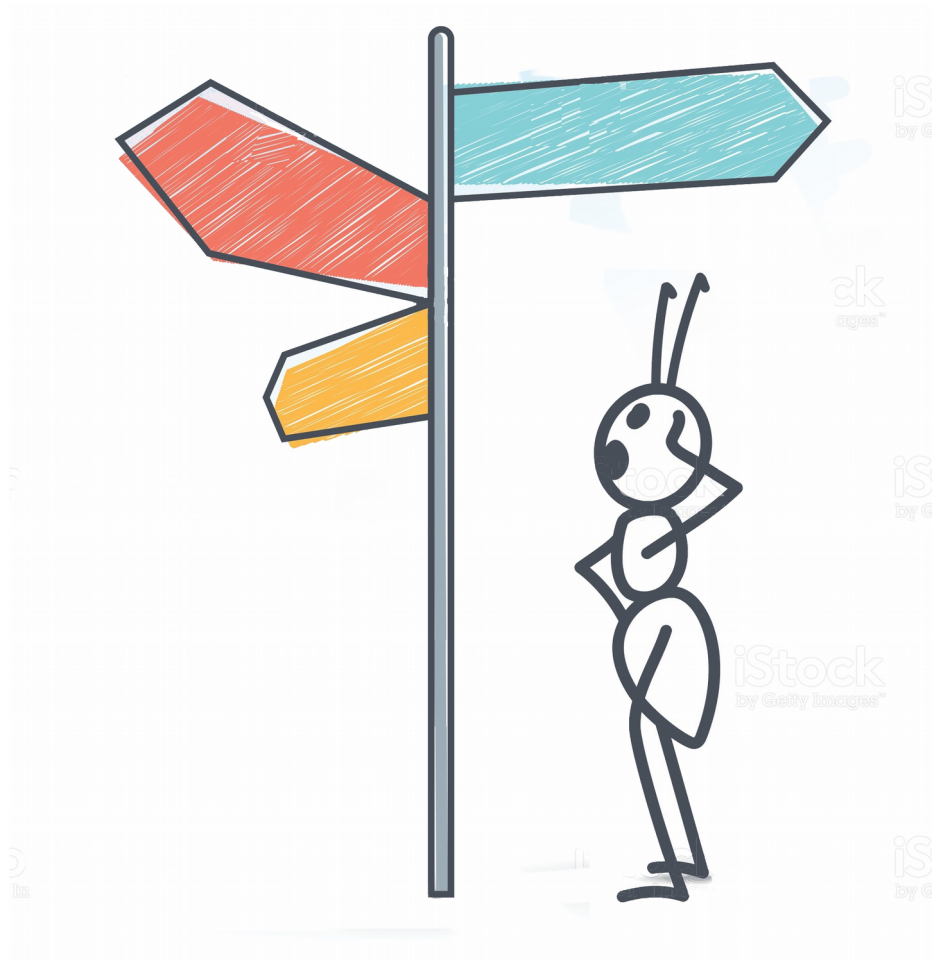


# ALGORITHMIQUE COMBINATOIRE:

## Voyageur de commerce



n:15603105

PLAVSIC && BARRETT

Vanja

n:14501697

Molly

18/12/2017

## 1)Contenu :

<u>1.1.1</u>	Enoncé du projet.....	3
<u>1.1.2</u>	Description des fonctions.....	3
<u>1.1.3</u>	Compilation et affichage .....	6
<u>1.1.4</u>	Difficultés.....	7
<u>1.1.5</u>	Source pour SDL.....	7

### 1.1.1)Enoncé du Projet: Voyageur de commerce

Notre projet est de voir le comportement des “fourmis” pendant leur parcours. Chaque ville contient une distance et une quantité de phéromones prédéterminés. On initialise deux fourmis par ville, chaque fourmi voyageant d’une ville à l’autre en laissant des phéromones sur son trajet pendant son voyage. Le choix de la ville se fera avec une probabilité de  $\frac{p_{ij}}{d_{ij}}$  (le taux de phéromones sur la ligne divisé par sa distance). Le nombre de phéromones va jouer sur le déplacement des autres fourmi, le but final étant de trouver la distance la plus courte (emprunté par une des fourmis) et voir le chemin le plus utilisé.

### 1.1.2)Description des fonctions :

Toutes nos fonction se trouvent dans un seul fichier (plus un Makefile):  
Pour faire notre parcours, nous avons d’abord eu besoin de créé deux classes:

La classe arrete contient :

- deux pair de float qui contient les coordonnées des deux points d’une arrete.
- deux float, une contenant la distance de l’arrete et l’autre celle des phéromones
- un int qui va stocker le nombre de passage quand une fourmi passera par l’arrete

La classe fourmi contient :

- un vecteur de int qui contient les villes visitées
- deux int, un qui stocke la place actuelle et l’autre la distance parcourue par la fourmi
- un booléen, qui renvoie true si la fourmi est la plus rapide (renvoie false dans le cas contraire).

Nous avons créé 3 typedefs :

```
typedef vector<fourmi> fourmisse;  
typedef vector<arrete> graph;  
typedef vector<float> vecf;
```

Les fonctions que nous avons créé pour accomplir ce parcours sont les suivantes:

-create\_vertex:

Notre fonction prend en paramètre un sommet et met dans un vecteur de pair de float des randoms, c'est la fonction qui génère des sommets aléatoires.

-create\_graph:

Cette fonction prend en paramètre un sommet et un vecteur de pair de float, et donne les valeur nécessaires pour créé notre graph (assigner les coordonnées, créer la ligne, assigner la distance pour chaque arrete et leur pheromones).

-fourmi\_distributeur :

Cette fonction prend en paramètre un sommet et renvoie un vecteur de fourmis. Elle donne a chaque fourmi sa place, la ville visitée, la distance parcourue, et si c'est la fourmi la plus rapide.

-seen :

Cette fonction vérifie si la fourmi est déjà passée par cette ville.

-somme\_proba :

Cette fonction calcule la probablité de toutes les arretes connectées à la ville.

-parcours\_fourmi:

Cette fonction est notre parcours de fourmi, c'est elle qui va faire bouger toutes les fourmis, modifier les phéromones, afficher la ligne la plus rapide et recommencer tout depuis le début après que les fourmis aient visiter toutes les villes.

Pour créer notre visualisation graphique on avait besoin de plusieurs fonctions de SDL. Les fonctions les plus **importantes** pour nous étaient la fonction ligne et cercle mais on avait besoin d'une autre fonction afin de les utiliser.

Les fonctions SDL :

-La fonction ligneHorizontale :

Fonction qui sert à créer des lignes horizontales.

-La fonction cercle :

Cette fonction nous sert à créer les cercles qui representent les "villes"

-La fonction echangerEntiers :

Fonction qui sert a swap deux int

-La fonction set Pixel:

Fonction qui sert à afficher les coordonnées x et y sur l'écran

-La fonction setPixelVerif:

Fonction qui vérifie l'emplacement des points (si elles sont aux bonnes coordonnées)

-La fonction ligne :

Cette fonction est très importante, elle récupère les coordonnées des deux points et la couleur que l'on désire utiliser

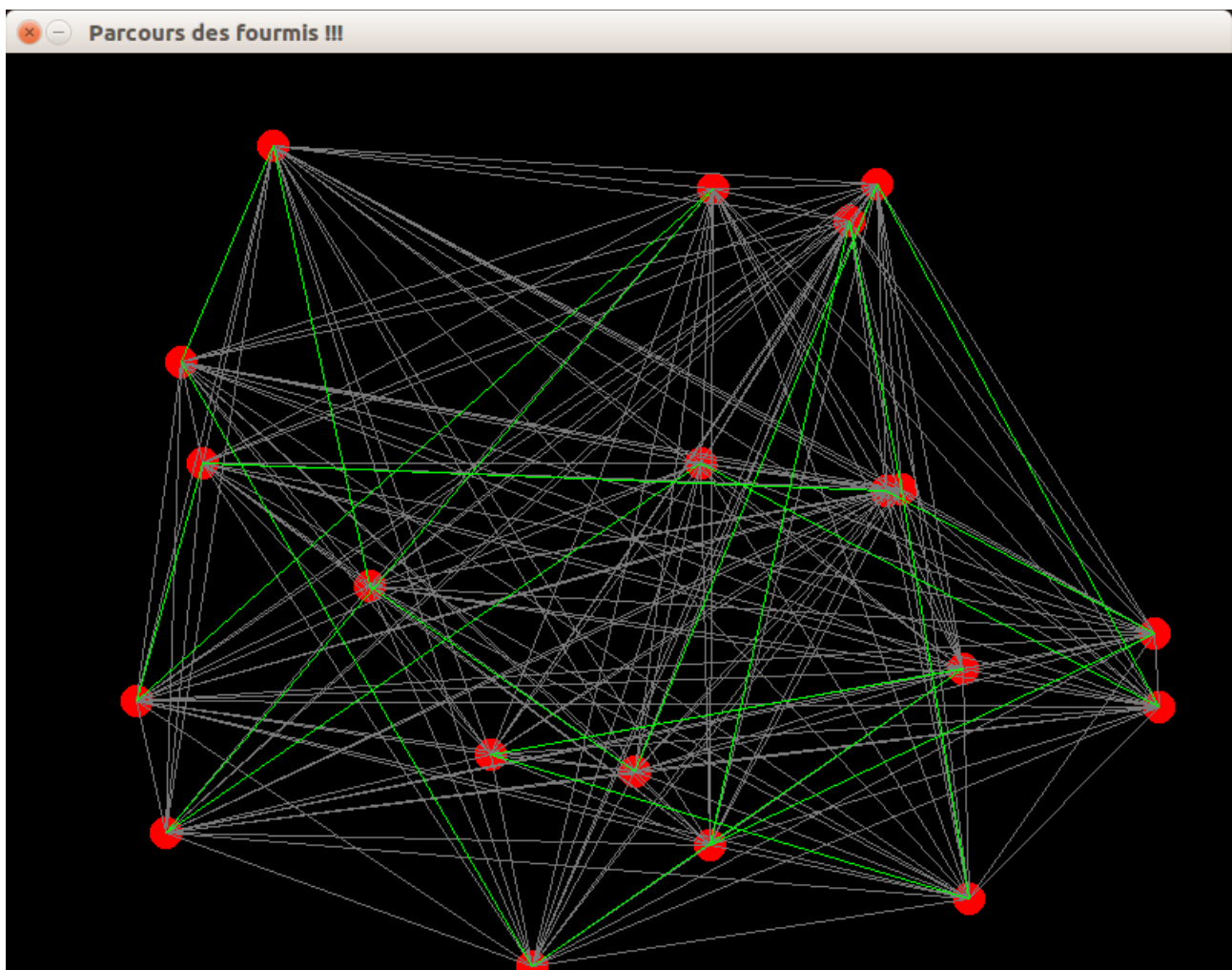
-La fonction InitSDL:

Fonction qui créer la fenêtre d'affichage

### 1.1.3)Compilation et affichage

Pour compiler, il faut suffir d'un make et d'exécuter le programme avec ./projet.

Pour quitter la visualisation graphique, il est impossible d'utiliser un Ctrl+c ou de cliquer sur la croix, il faut tuer le processus (à prendre en compte lors de l'exécution du code).



Les cercles rouges représentent les villes, les lignes vertes le parcours de la fourmi la plus rapide, et les lignes grises la connexion entre chaque ville.

#### 1.1.4)Difficultes

La plus grande difficulté que nous ayons eue est avec l'affichage des phéromones. On ne peut pas bien voir le changement de couleurs de la ligne à chaque passage de la fourmi, celle-ci étant beaucoup trop rapide (la ligne est toujours présente dans le code mais en commentaire).

#### 1.1.4)Source pour SDL

<http://anomaly.developpez.com/tutoriel/sdl/partie2/>