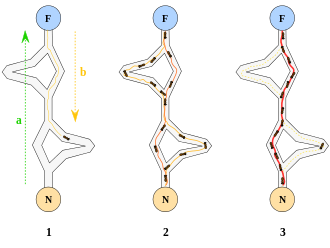


Ecole Nationale supérieure de l’enseignement TechniqueUniversité Hassan II MohammediaDépartement Mathématique Informatique



**Elaboré par :**

Yasser NADIR

**Encadré par :**

Mr Abdelwahab Naji

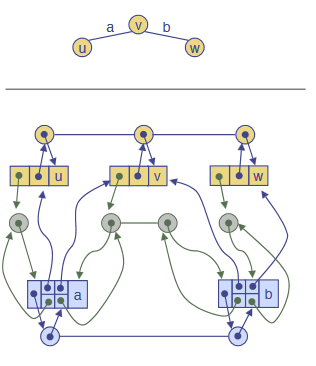
**Année universitaire**2020-2021

Sommaire :

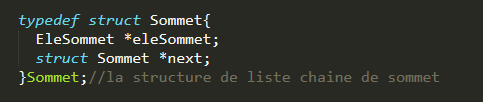
1. **La présentation des structures des données**
   1. Les structures de graphe
      * La structure des sommets
      * La structure des arcs
   2. Les structures de la fourmi
      * La structure de la fourmi
      * La structure de chemin
2. **Les fonctions de manipulations des structures de données** 
   1. Les fonctions qui gèrent les sommets
      * Création
        1. Elément sommet
        2. Sommet
      * Ajout
   2. Les fonctions qui gèrent les arcs
      * Création
        1. Elément arc
        2. Arc
      * Ajout
   3. Les fonctions qui gèrent les graphes
      * Création
      * Initialisation
      * Ajout de sommet
      * Ajout d’arc
   4. Les fonctions qui gèrent les chemins
      * Création
      * Ajout
   5. Les fonctions qui gèrent les fourmis
      * Création
      * Ajout
3. **Le scénario N°1 : deux fourmis avec des chemin déjà fixer**
4. **Le scénario N°2 : Nombre ‘m’des fourmis circulent de façon libre**
5. **La visualisation avec python et tkinter**

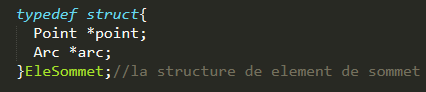
**La présentation des structures des données**

* 1. La structure de graphe

Inspirant de cette modalisation des graphes importer de <https://www.iro.umontreal.ca/~hamelsyl/grapheA09_4.pdf>, j’ai réussi à créer la structure de données de graphe baser sur liste chainée de sommet et arc et un genre de liaison entre les sommets et les arcs

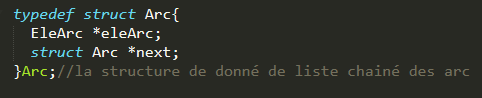
La structure des sommets :

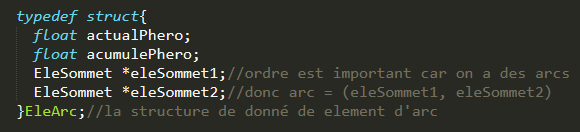
La structure de sommet est une liste chainée des élément sommet



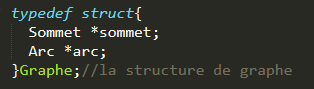
La structure d’élément de sommet se construit du point, et une liste chainée des arcs, la spécifié de ces arcs se construit de ce sommet et un autre sommet, cette liaison représente les sommets adjacents à ce sommet

La structure des arcs :

La structure d’arc est une liste chainée des éléments d’arc

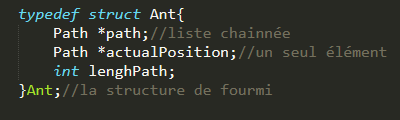


La structure d’élément d’arc se construit de la quantité de phéromone accumulé et la quantité de phéromone se trouve dans l’instant t un pointeur vers 2 éléments de sommet qui construit l’arc

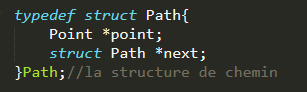
Pour graphe il est représenté comme la suite :  un pointeur de sommet et un pointeur d’arc

1. Les structures de la fourmi

La structure de la fourmi :

La structure de fourmi se construit de liste chainée de chemin(pat) un pointeur vers emplacement actuel (actualPosition) et la longueur du chemin (lenghPath)

La structure de chemin :

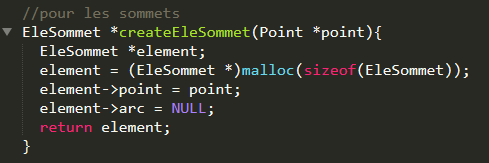
La structure de chemin (path) se construit de point et un pointeur vers la case suivante (liste chainée)

**Les fonctions de manipulations des structures de données**

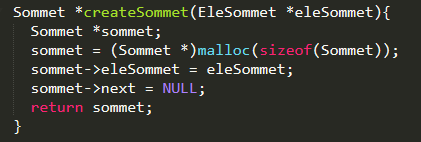
1. Les fonctions qui gèrent les sommets

* Création

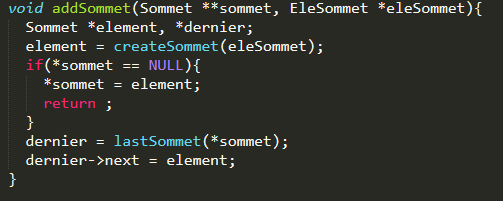
Elément sommet :



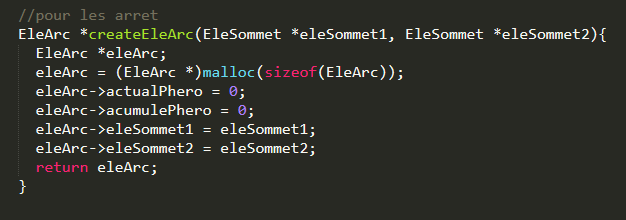
Sommet :



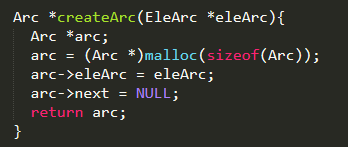
Ajout :



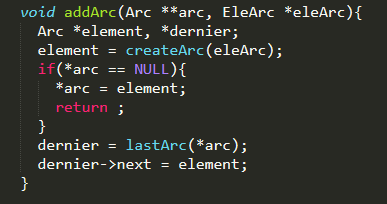
1. Les fonctions qui gèrent les arcs
   * + Création
       1. Elément arc



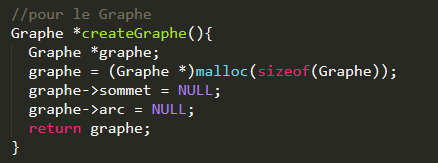
* + - 1. Arc



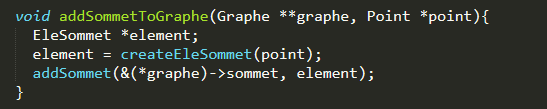
* + - Ajout

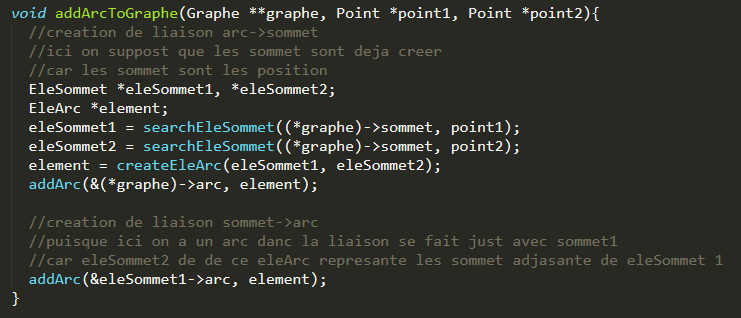


1. Les fonctions qui gèrent les graphes
   * + Création



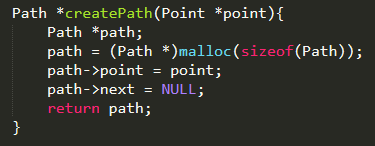
* + - Ajout de sommet



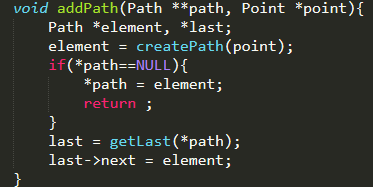
* + - Ajout d’arc
    - Initialisation



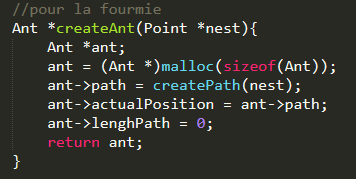
1. Les fonctions qui gèrent les chemins
   * + Création



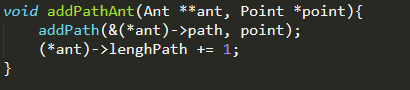
* + - Ajout



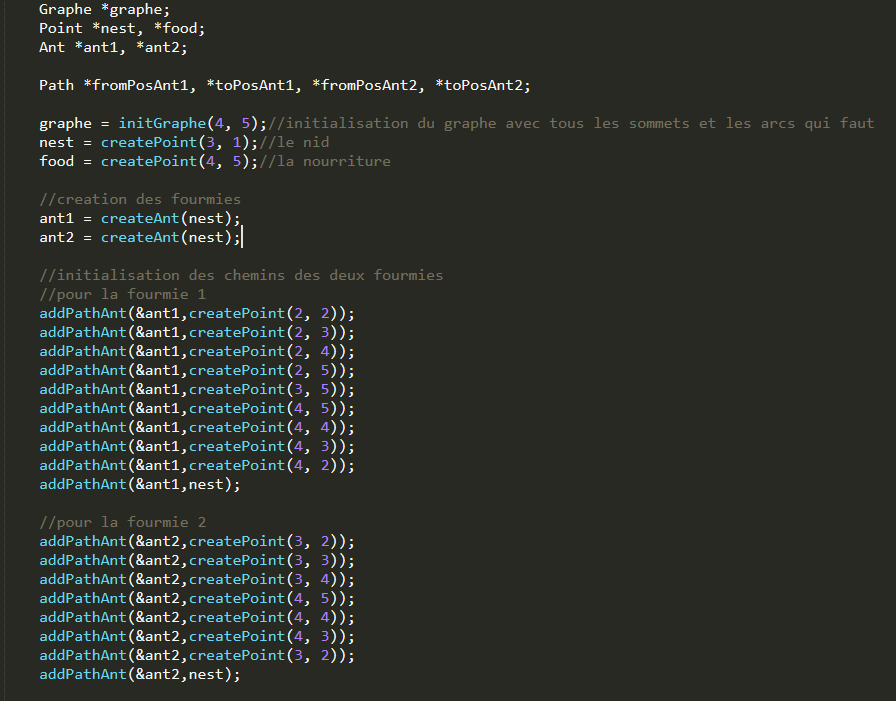
1. Les fonctions qui gèrent les fourmis
   * + Création

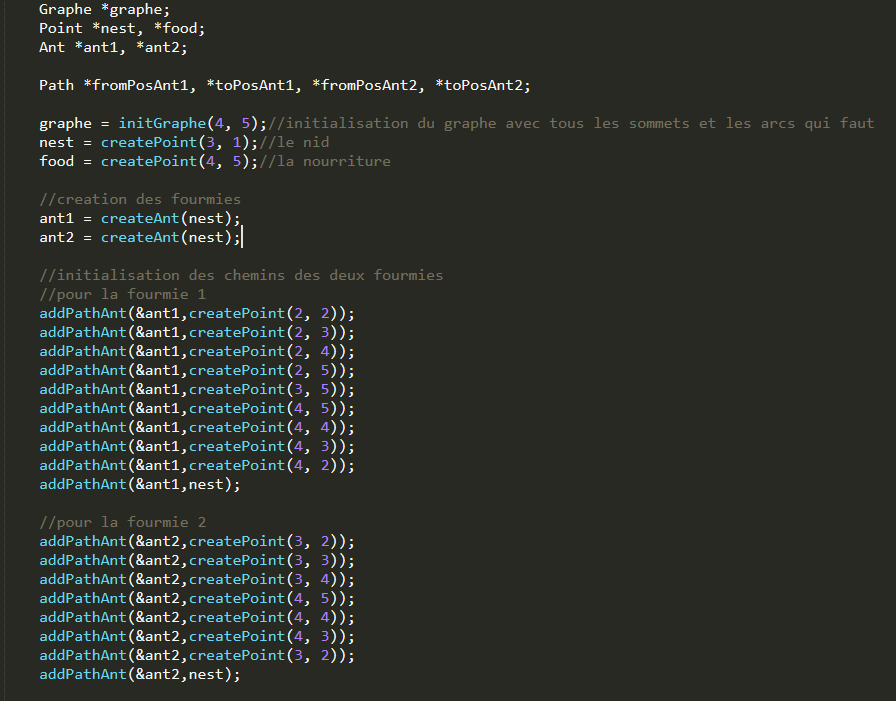


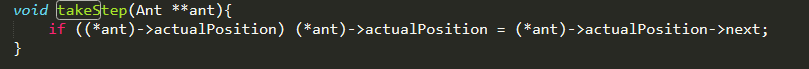
* + - Ajout

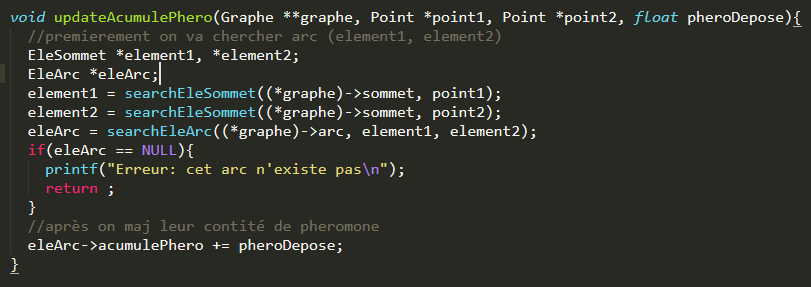


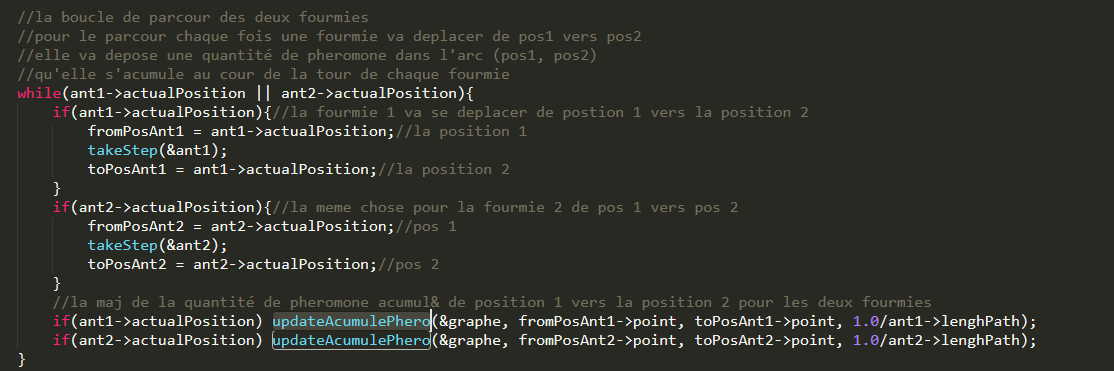
**Le scénario N°1 : deux fourmis avec des chemin déjà fixer**

* + - Initialisation des deux fourmis, du graphe, nid et la nourriture
    - Initialisation des chemins des deux fourmis



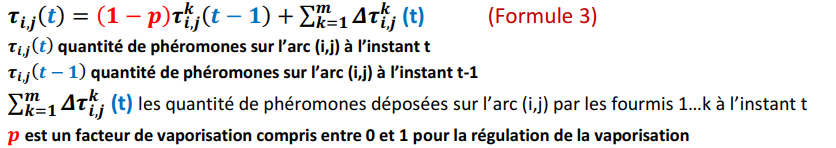
* Le parcours de la fourmi : chaque fourmi est caractérisée par 2 pointeurs path et actualPosition, path représente le chemin que la fourmi va parcourir, et actualPosition va pointer sur la position actuelle de la fourmi, et on plus des 2 pointeur la fourmi ou ‘ant’ se construit de lenghPath qui représente la longueur du chemin pour le parcours, on a défini une ‘takeStep’ qui va décaler le pointeur actualPosition puisqu’il n’est pas NULL
* Accumulation de la phéromone :

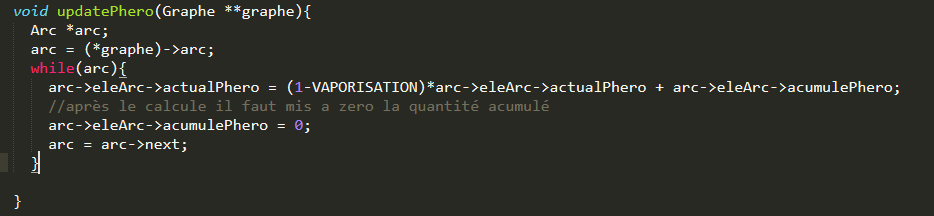
Quand la fourmi suit son chemin elle laisse une quantité de phéromone sur l’arc qu’elle a parcouru, pour cela on a défini la fonction ‘updateAcumulePhero’ qui accumule la quantité de phéromone dépose par la fourmi dans un arc

Donc pour que la fourmi parcourt dans le graphe, on suggère le scenario suivant :

On prend la position avant le déplacement et après le déplacement, ces deux positions vas nous aider à accumuler la quantité de phéromone dans l’arc.

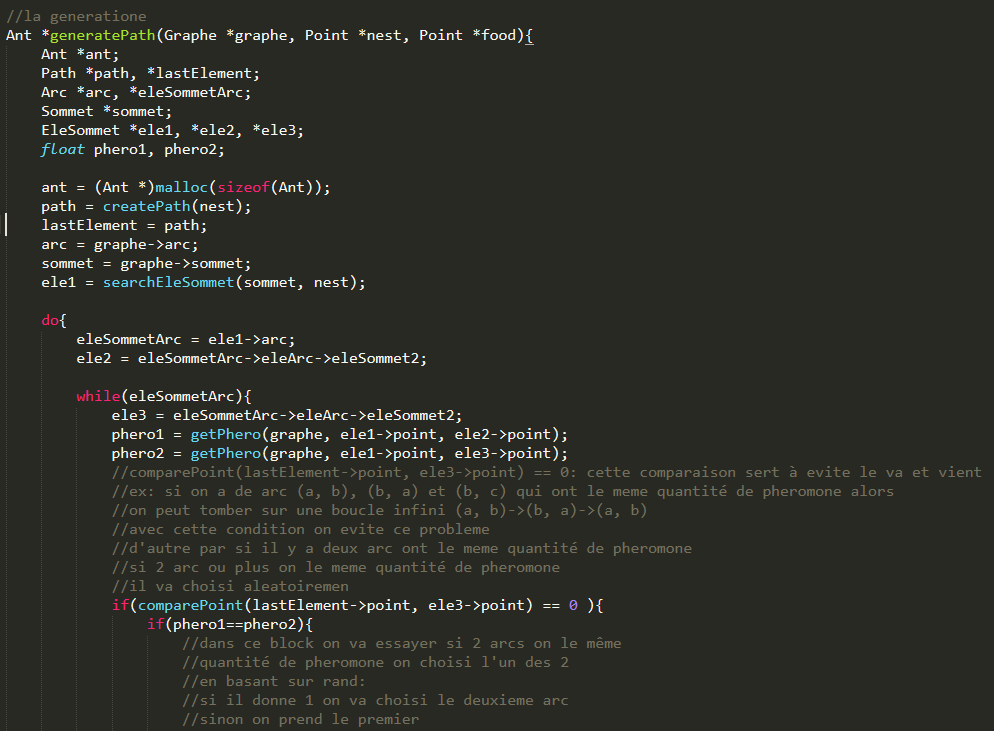
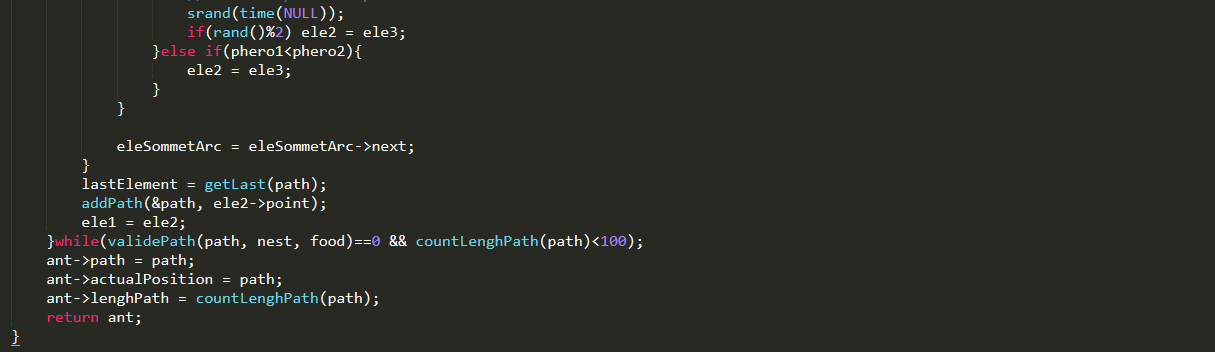
* La mise à jour de la quantité de phéromone

La mise à jour de la quantité de phéromone se fait par le parcours dans tous les arcs du graphe et appliquer la formule suivante

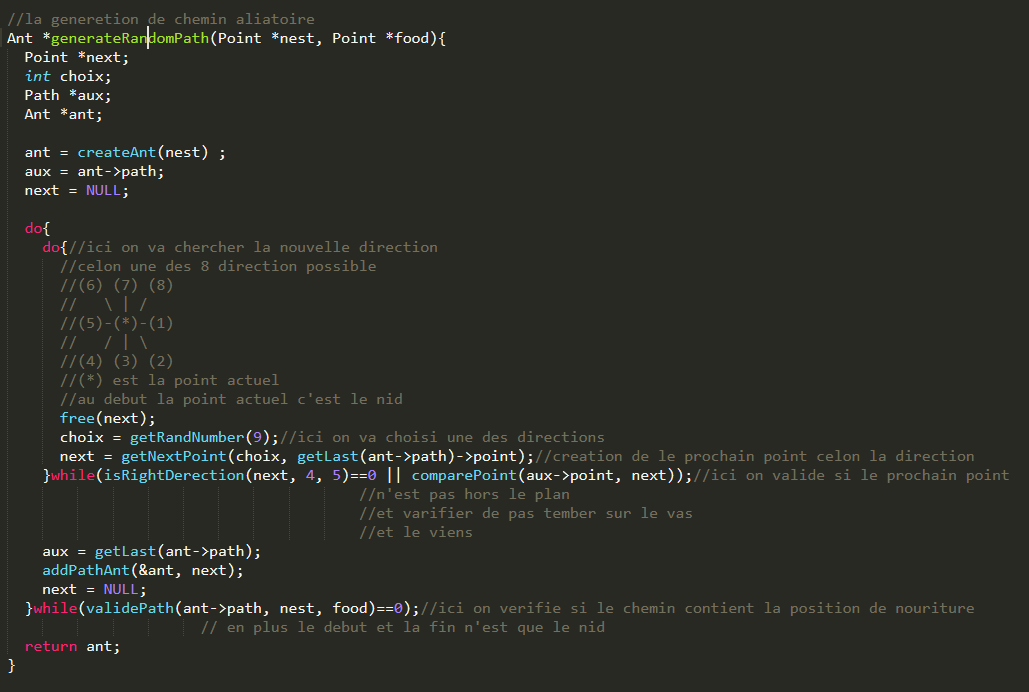
Le facteur de vaporisation est représenté par une constante ‘#define VAPORISATION 0.3’

* La troisième fourmi et la recherche du chemin

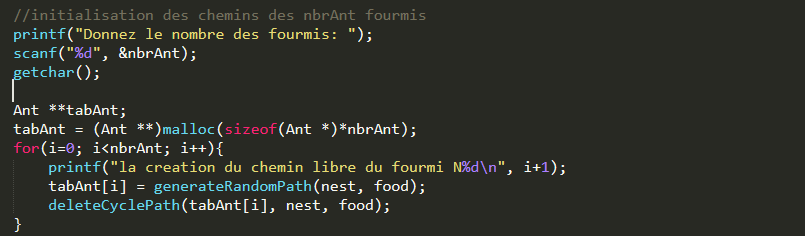
Après la définition de la troisième fourmi comme la première et la deuxième, on va utiliser la fonction ‘generatePath’, cette fonction va parcourir tous les arcs du graphe, par la suite il va choisir les arcs qui ont la quantité de phéromone la plus haute, si elle trouve 2 ou plusieurs arcs ont la même quantité de phéromone, elle va choisir aléatoirement l’un des arcs. Comme une condition d’arrêt on a testé si le chemin est valide, pour qu’un chemin peut être un chemin valide il faut contient emplacement de nourriture, commence et fini par le nid, et la longueur du chemin ne dépasse pas 100, voyant que la langueur de plus court chemin est 8



**Le scénario N°2 : Nombre ‘m’des fourmis circulent de façon libre**

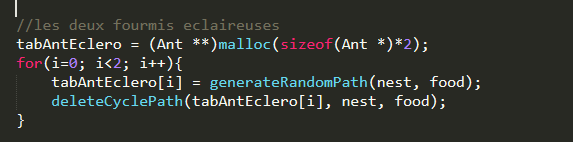
* Le chemin libre : pour que les fourmis parcourent de façon libre, j’ai proposé de générer un chemin en choisissant des directions aléatoires, en vérifiant toujours si le chemin généré un chemin validé
* Plus que 2 fourmis :

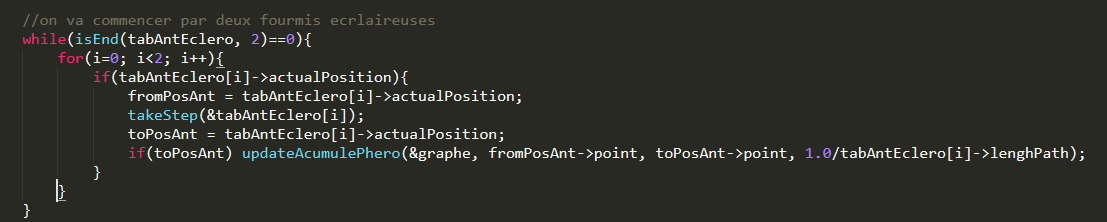
Pour donner la possibilité à plus que 2 fourmis, on crée un tableau des doubles pointeurs de type ‘ant’, le nombre des fourmis est défini par l’utilisateur



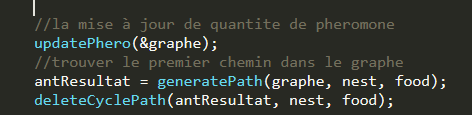
* 2 fourmis éclaireuses

En plus du tableau des fourmis on crée 2 fourmis éclaireuses pour commencer la recherche



Et le parcours de 2 fourmis éclaireuse :

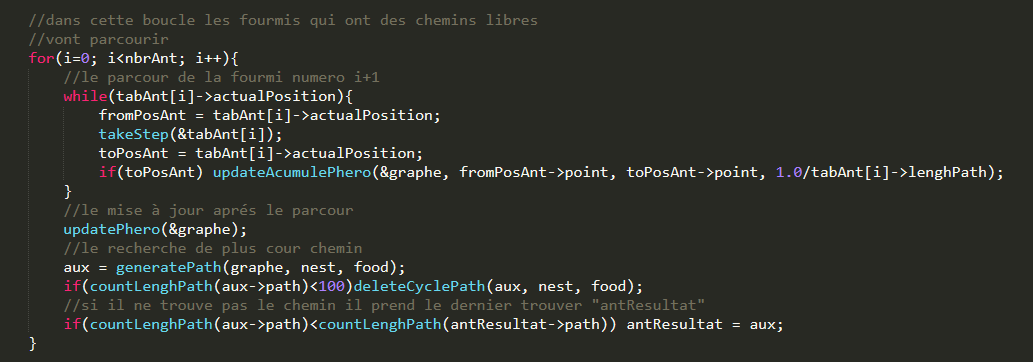
Après le parcours on mise à jour la quantité de phéromone, et on cherche sur le plus court chemin :



La fonction deleteCyclePath est un moyen d’optimisation parce que parfois les chemins générés aléatoirement peut contient des cycles, et ça peut ne pas aider à trouver le plus court chemin

* Le parcours de m fourmis :

Chaque fourmi parcourt, après chaque parcours on mit à jours la quantité de phéromone après on cherche un chemin, si le chemin trouvé ‘aux’ est plus cours qu’ancien ‘antResultat’ on va pointer sur le nouveau chemin



**La visualisation avec python et tkinter**

J’ai choisi python pour visualiser le parcours des fourmis avec la bibliothèque tkinter, premièrement on va écrire tous les chemin parcouru (sauf le chemin des fourmis éclaireuse) dans un fichier txt par la suite python va lire se fichier et parcourir les fourmis

C’est les fourmis qui parcourent

C’est la fourmi qui parcourt le plus court chemin