 **Universite Sultan Moulay Slimane Faculte polydisciplinaire de beni mellaL Departement mathematiques et informatique**

Universite sultan moulay slimane

faculte polydisciplinaire de beni meellal

departement matheatique et informatique

***MINI PROJET : ALGORITHME DIJKSTRA***

**Présenté par :**

*MOHAMED ER -RAGUIG*

*MINA AYYAD* **Encadre par** :

***Pr . Sadqi Yassine***

**Année Universitaire :**

**2023- 2024**

* d

Plan

* Introduction
* Problématique
* Les graphes
* Algorithmes Dijkstra
* Les solutions proposées
* Conclusion

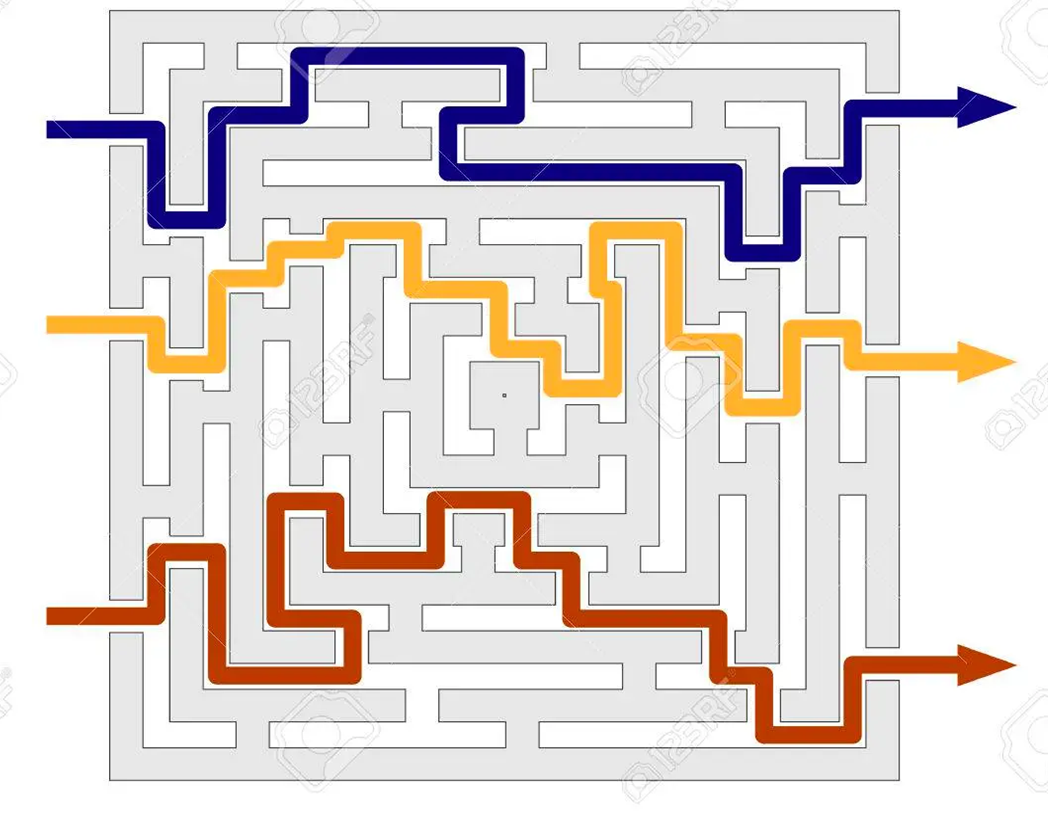
Introduction

L'algorithme de Dijkstra est une méthode de calcul utilisée pour déterminer le chemin le plus court entre des nœuds dans un graphe pondéré. Proposé par l'informaticien néerlandais Edsger Dijkstra en 1956, cet algorithme est essentiel dans de nombreux domaines, allant de la planification des itinéraires dans les réseaux de transport à l'optimisation des réseaux informatiques.

Dans le cadre de ce mini-projet, nous explorerons les fondements théoriques de l'algorithme de Dijkstra, sa mise en œuvre pratique, ainsi que ses applications dans divers contextes. Ce projet vise à fournir une compréhension approfondie de la manière dont cet algorithme fonctionne et comment il peut être utilisé pour résoudre des problèmes réels.

Problématique

Quel est le chemin le plus court pour se déplacer de l’entrée a la sortie du labyrinthe ?

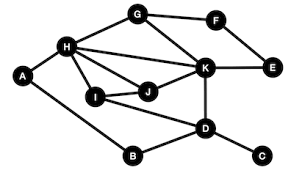


Les graphes

* Définitions
* Un [**graphe**](https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/)est une structure de données non linéaire constituée de **nœuds** (ou sommets) et **d'arêtes** . les arêtes sont des lignes ou des arcs qui relient un paire de nœuds quelconques du graphe*.*
* Deux sommets sont **adjacents** s’ils sont reliés par une arête.
* **Ordre d’un graphe**est le nombre total de sommets
* **Une boucle** est une arête reliant le même sommet.
* Un graphe est **simple** s’il ne comporte aucune boucle et que deux arêtes ne relient jamais la même paire de sommets.
* **Le degré** d’un sommet est égal au nombre d’arêtes qui le relient aux autres sommets..
* **Propriétés des graphes :**
* **Un graphe orienté** est un graphe dont les arêtes sont orientées. Elles ont une origine et une extrémité.
* **La matrice associée** de n lignes et n colonnes est la matrice où le terme à l’intersection de la ie ligne et de la je colonne est égal au nombre d’arêtes reliant les sommets i et j.

**Algorithme Dijkstra**

Il y on a plusieurs méthodes pour implémenter un algorithme Dijkstra en langage de programmation C , celles-ci dépendent de la représentation d’un graphe et les fonctions de structures de données à employer.



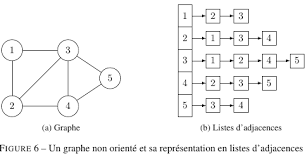
**Solutions proposées**

Le procédure de la recherche de plus court chemin peut être basé sur :

1. Les listes d’adjacence
2. Les tas binaire
3. Les listes chainées
4. Les tables d’hachage

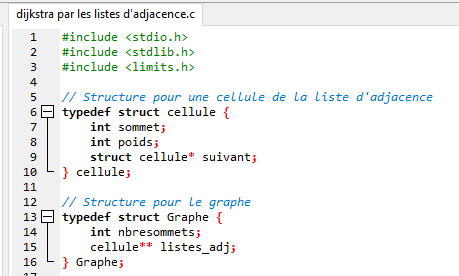
**Algorithme Dijkstra: Listes d’adjacence**

* **Définitions**
* une liste d'adjacence est une structure de données utilisée pour représenter un graphe.
* La liste d'adjacence d'un graphe non orienté, est la liste des voisins de chaque sommet. Et pour un graphe orienté la liste d’adjacence , pour chaque sommet, est la liste de nœuds à la tête de chaque arête ayant le sommet comme queue

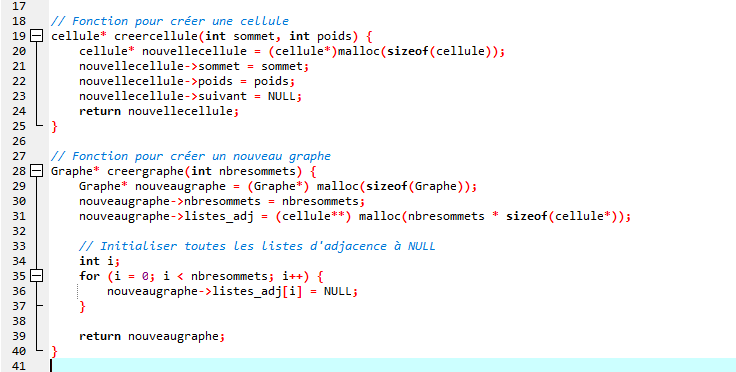


**Représentation de la liste adjacence en C et implémentation de l'algorithme de Dijkstra .**

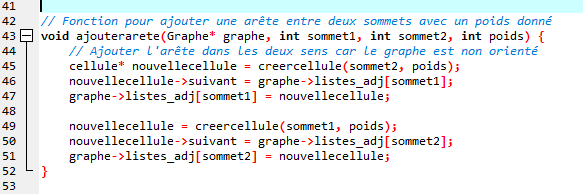
* ***Etape 1:*** créer une structure de donnée pour modéliser une cellule de la liste d’adjacence et une autre pour modéliser un graphe.



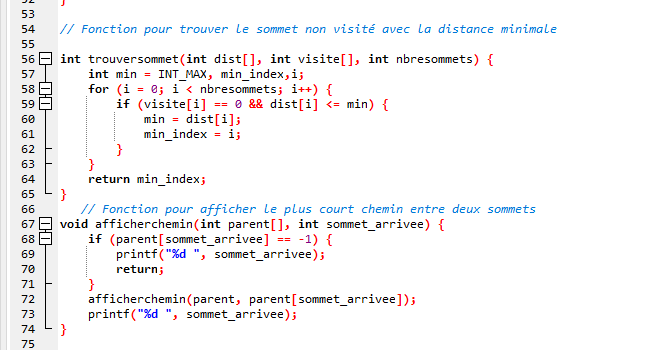
* ***Etape 2:*** innover deux fonctions qui permettent de créer à chaque fois une nouvelle cellule ou un nouveau graphe.



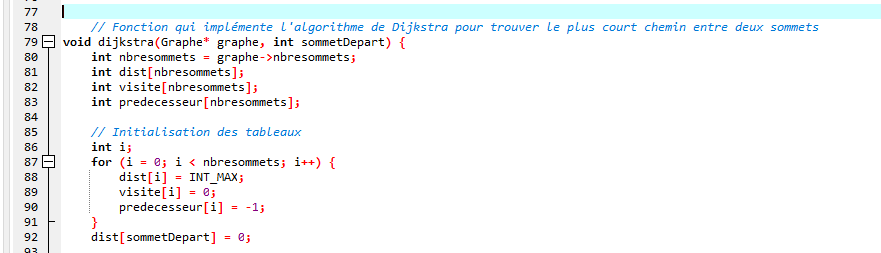
* ***Etape 3:*** ajouter un arrêt entre deux sommet avec un poids choisi dans le cas d’un graphe non orienté



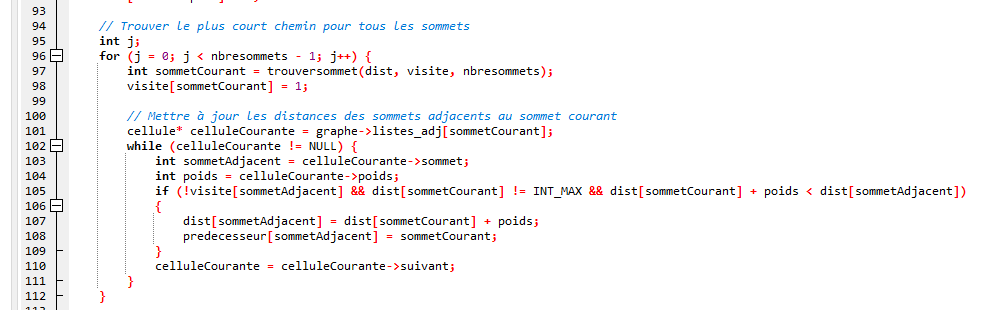
* ***Etape 4:*** chercher le sommet voisin non visité et afficher le plus court chemin entre deux sommets du graphe



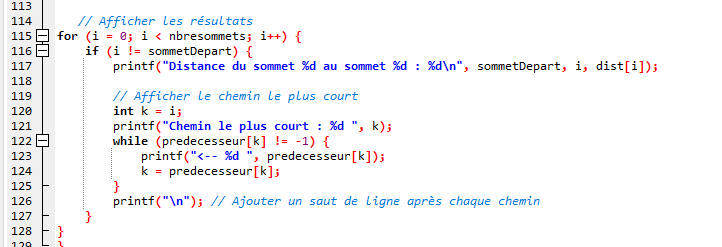
* ***Etape 5:*** implémenter l’algorithme Dijkstra pour trouver le chemin le plus court entre deux sommet donnés



* ***Etape 6:*** remplir le tableau sommet puis metter à jour les distances des sommet au sommet courant



* ***Etape 7***: retourner le résultat de l’algorithme Dijkstra à l’aide des listes d’adjacences en affichant le plus court chemin entre les deux sommets précis

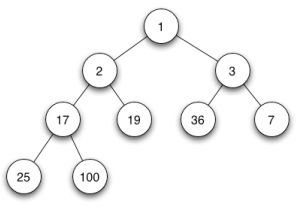
******

La complexité de ce programmes est : en notons n le nombre do sommets dans le graphes , est :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **creecellule** | **creegraphe** | **ajouterarete** | **trouversommet** | **afficherchemen** | **dijkstra** |
| **O(1)** | O(n) |  |  |  |  |

**Algorithme Dijkstra: Les tas binaires**

* **Définitions :**
* Un tas binaire est une structure de données permettant de modéliser une file à priorités.

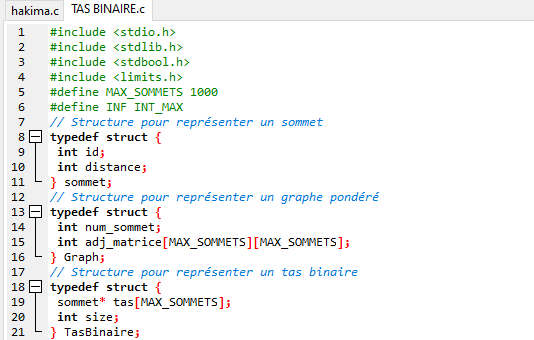


* Cette structure de données maintient un ordre partiel sur les données via un arbre binaire. Cet arbre respecte les propriétés suivant
* chaque nœud a une priorité supérieure à celles de ses enfants
* un nouveau niveau n’est créé que si le niveau précédent est plein ;
* les niveaux de l’arbre sont remplis de gauche à droite.

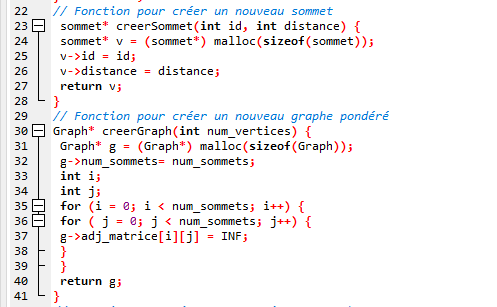
* **Représentation des tas binaires en C et implémentation de l'algorithme de Dijkstra .**

Pour implémenter l’algorithme Dijkstra à l’aide des les tas binaires ont suit les étapes suivants :

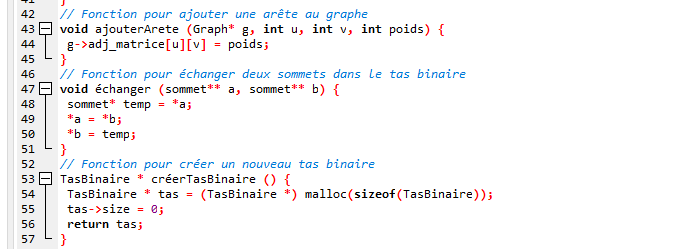
* ***Etape 1:*** déclarer trois de structures de données pour représenter **les sommets** , **le graphe pondère** et une autre pour **le tas binaire**



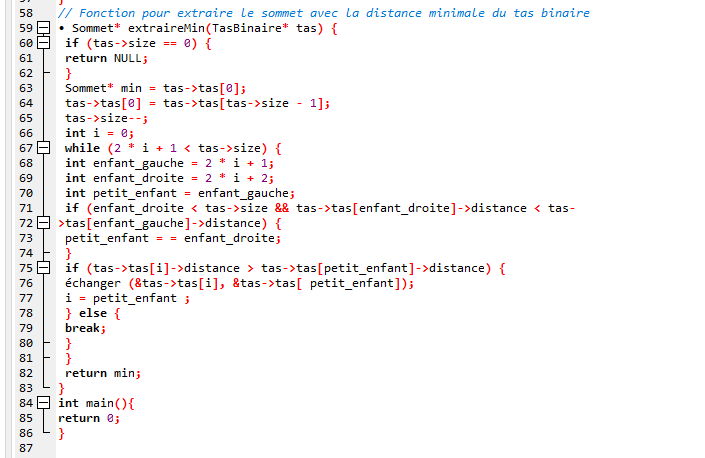
* ***Etape 2:*** implémenter deux fonctions permettent de créer a chaque fois un noiveau sommet un nouveau graphe ;



* ***Etape 3:*** implémenter trois fonctions qui serrent à ajouter une arête au graphe et échanger les sommets dans un tas binaire ainsi créer un nouveaux tas binaire en cas de besoin :



* ***Etape 4:*** en fin cette fonction return le sommet suivant avec la distance minimal :



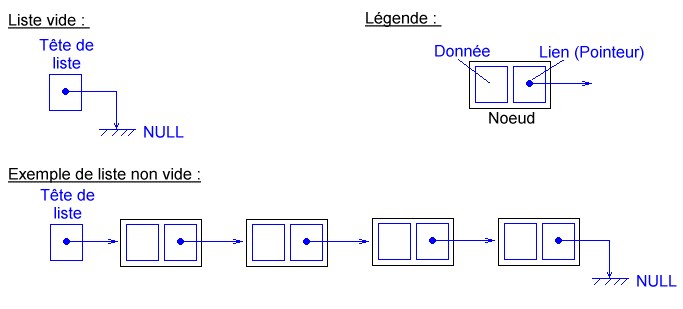
La complexité de ce programmes ,en notons n le nombre de sommet du graphe , est :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **creersommet** | **creergraphe** | **ajouteraAret** | **echanger** | **creerTasBiniare** | **extraireMin** |

**Algorithme Dijkstra :Les listes chainées**

* **Définitions :**

Une liste simplement chaînée est une structure de données à l’intérieure de laquelle les objets sont ordonnés de façon linéaire



* **Propriétés de base :** 
  + Une liste chaînée est accessible uniquement par sa tête .
  + Une liste chaînée est constituée de 0 (liste vide) ou n cellules (maillons).
  + Sans la tête impossible de savoir où commence la chaine et sans la queue impossible de savoir où elle s'arrête .

Analyse du Projet :

1. ANALYSE DE MARCHE

L'analyse de marché vise à comprendre les caractéristiques, les besoins et les tendances d'un marché spécifique pour un produit ou un service. Elle est essentielle pour déterminer la faisabilité commerciale d'un projet.

1. ANALYSE TECHNIQUE

L'analyse technique évalue les aspects pratiques et les exigences technologiques nécessaires pour réaliser un projet. Elle examine les ressources, les infrastructures et les technologies requises.

1. ANALYSE FINANCIERE

L'analyse financière évalue la viabilité financière du projet. Elle examine les coûts, les revenus attendus, la rentabilité et les risques financiers.

1. ANALYSE ECONOMIQUE

L'analyse économique évalue l'impact global du projet sur l'économie, incluant les aspects macroéconomiques et microéconomiques.

1. ANALYSE ECOLOQIQUE

L'analyse écologique évalue l'impact environnemental du projet et cherche à promouvoir des pratiques durables et respectueuses de l'environnement.

Tâches

Objectifs de management

Plannings



Objectifs du projet

Plan de mise en œuvre

Objectifs du client

Ressources

**LOREM IPSUM**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor.

5 980

**LOREM IPSUM**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor.

-1,19

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor.

113 200 50 €

CONCLUSION

L'algorithme de Dijkstra est un outil puissant et polyvalent qui joue un rôle crucial dans la résolution de nombreux problèmes de recherche de chemin. En étudiant cet algorithme, nous approfondirons non seulement nos compétences en programmation et en algorithmique, mais aussi notre capacité à aborder des problèmes complexes de manière systématique et efficace. Ce mini-projet constitue une opportunité précieuse pour appliquer des concepts théoriques à des situations pratiques, et pour développer une expertise précieuse dans le domaine de l'informatique et de l'ingénieurie

MERCI POUR VOTRE ATTENTION