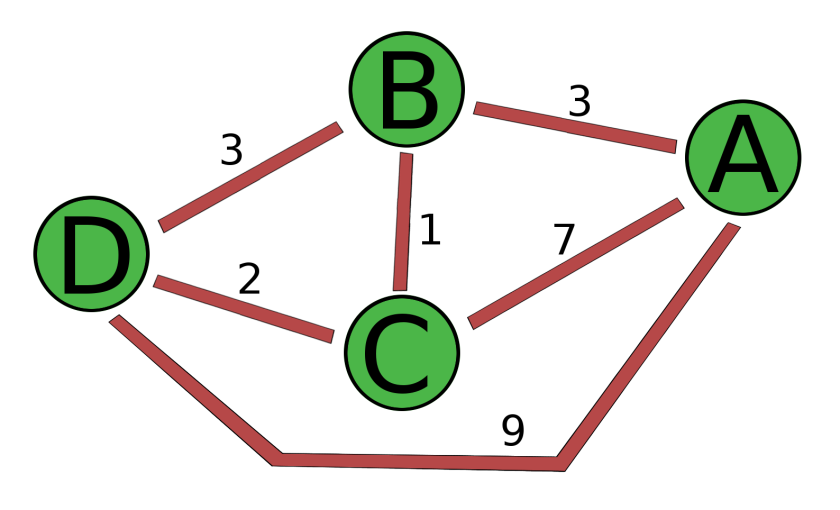
**Université Ibn Tofail**

****

**Pr.Khalid OUNACHAD**

Réalisé par :

**JIYDH Abdulmajeed**

**Le plus court chemin avec python(PERT)**

Contexte général

* Methode PERT

La méthode PERT est une méthode de gestion de projet visant à prévoir les propriétés d’un projet en termes de temps délais et couts.

Son principe est de découper un projet en un ensemble d’actions appelées taches et de les représenter sous forme graphique selon un graphe de dépendances.

Grâce à la chronologie et l’interdépendance de chacune des taches, on structure ainsi l’ensemble du projet et on peut alors planifier la réalisation de chacune des taches les unes par rapport aux autres afin de minimiser les délais, ainsi que réduire l’impact des retards lors de l’exécution des différences tâches.

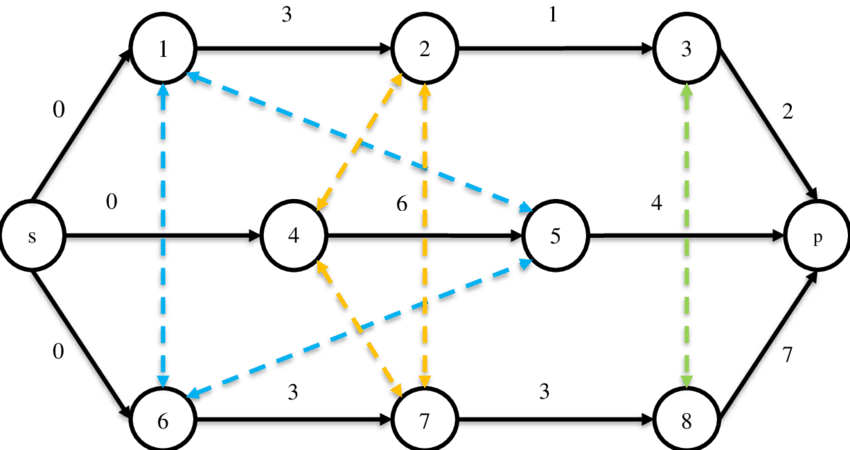
* L'algorithme de Dijkstra

L'algorithme de Dijkstra est un algorithme de recherche de chemin, comme ceux utilisés dans le routage et la navigation.

Nous allons l'utiliser pour trouver le chemin le plus court entre deux nœuds dans un graphe.

Objectif et principe du sujet

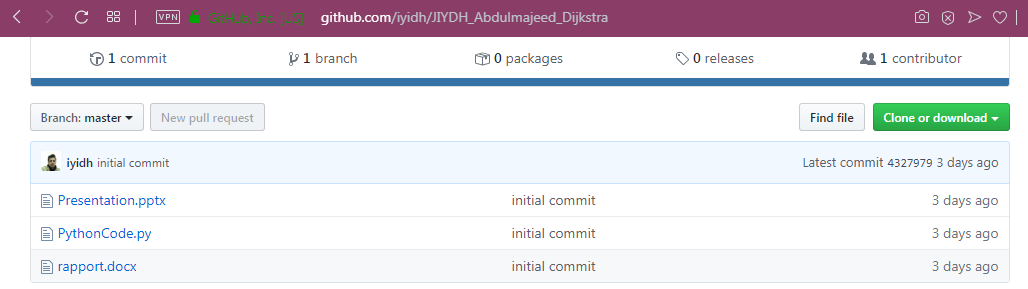
L'algorithme de Dijkstra sert à résoudre le problème du plus court chemin. Il permet, par exemple, de déterminer un plus court chemin pour se rendre d'une ville à une autre connaissant le réseau routier d'une région. Plus précisément, il calcule des plus courts chemins à partir d'une source dans un graphe orienté pondéré par des réels positifs. On peut aussi l'utiliser pour calculer un plus court chemin entre un sommet de départ et un sommet d'arrivée.

Puisque nous avons parlé d'un algorithme, l'objectif principal de notre sujet est de trouver le plus court chemin entre le point de départ et le point d'arrivée.

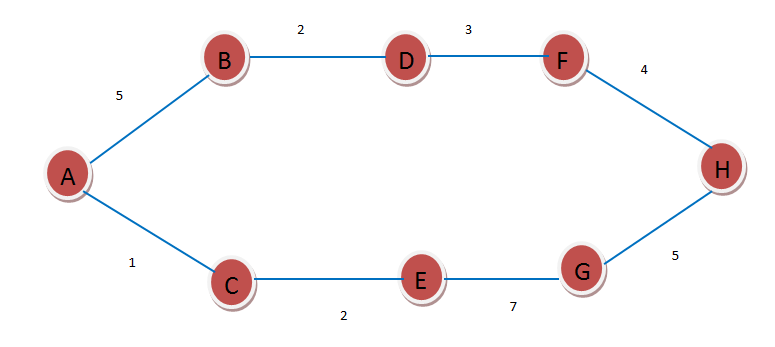
Guide d’utilisation et (URL)

* Lien github de l’application

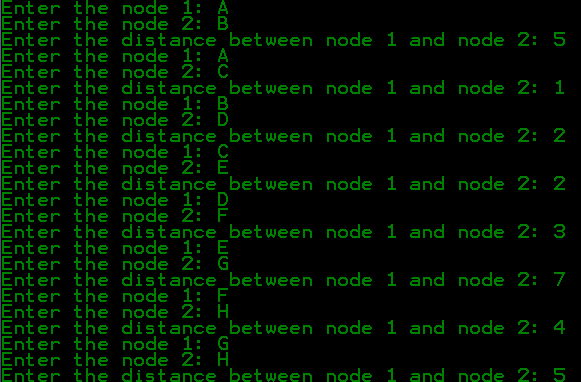
<https://github.com/iyidh/JIYDH_Abdulmajeed_Dijkstra>



Nous utilisons la figure ci-dessous pour tester notre application, Nous allons convertir le graphe suivant en cette forme. Par exemple, si nous choisissons les nœuds A et B, nous les formaterons comme suit **A==B==5** afin de faciliter le processus de saisir des données dans notre programme, et ainsi pour les autres nœuds.



Apres d’extraire les nœuds et les distances du graphe, nous allons saisir les nœuds et les distances, Nous entrerons le premier nœud puis le nœud suivant à partir du point de départ jusqu'à ce que nous atteignions le point final.



cp10.PNG

Conclusion et perspectives

Open Source désignant un logiciel pour lequel le code source d'origine est mis à disposition gratuitement et peut être redistribué et modifié, l'algorithme Dijkstra peut également être redistribué et modifié.

Je viens donc d'utiliser un algorithme existant et de le modifier. Par exemple, j'ai ajouté la fonctionnalité d'entrée à l'algorithme qui permet à l'utilisateur d'entrer les nœuds et la distance entre eux.

Il existe de nombreuses fonctions qui peuvent être ajoutées à cet algorithme. Par exemple, nous créons une interface qui permet à l'utilisateur d'entrer des nœuds et les distances entre eux et transformons cette interface en Set Up que tout le monde peut utiliser et télécharger via Internet.