Initialisation :

On charge le graphe du txt dans un tableau, exemple avec le graphe de test 6 :

|  |  |  |
| --- | --- | --- |
| 0 | 1 | 1 |
| 0 | 4 | 2 |
| 0 | 8 | 3 |
| 0 | 13 | 4 |
| 1 | 2 | 2 |
| 1 | 6 | 3 |
| 1 | 11 | 4 |
| 2 | 3 | 3 |
| 2 | 8 | 4 |
| 3 | 4 | 4 |

On vérifie si on peut effectuer Djikstra : recherche d’un arc de valeur négative.

On demande à l’utilisateur de quel sommet il souhaite partir (admettons qu’ici ce soit 0).

On crée un tableau avec n + 1 colonnes :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CC | 0 | 1 | 2 | 3 | 4 |

On parcourt le tableau du graphe en cherchant les arcs qui partent du sommet choisi par l’utilisateur précédemment.

Si c’est un 0 on prend la valeur de l’arc dans la 2ème colonne, puis la valeur du somment vers lequel est orienté l’arc, ici 1 -> on place donc dans le tableau de l’algo à la colonne d’index 2 la valeur de l’arc 1. On fait ça pour tous les arcs partant de 0 ce qui remplit en partie la 2ème ligne du coup, pour la case de 0 et lui-même on met 0 et dans celle où il n’y a pas de valeur on met un ∞.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CC | 0 | 1 | 2 | 3 | 4 |
| 0 | 0(0) | 1(0) | 4(0) | 8(0) | 4(0) |

On choisit celui vers lequel la valeur est la plus basse, ici 1.

On parcourt le tableau du graphe en cherchant les arcs partant de 1. Attention à ne pas prendre d’arc allant vers un sommet déjà présent dans le chemin le plus court.

On ajoute à chaque fois la valeur de l’arc avec celle de 1 (le sommet que l’on vient d’ajouter au chemin le plus court). On fait le minimum des valeurs trouvées avec celles déjà présentes dans le tableau, par exemple le min entre 4(0) et 1(0) +2(1) pour le sommet 2. S’il n’y en avait pas on rajoute simplement la valeur.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CC | 0 | 1 | 2 | 3 | 4 |
| 0 | 0(0) | 1(0) | 4(0) | 8(0) | 4(0) |
| 0,1 | . | . | 3(1) | 7(1) | 4(0) |

On choisit à nouveau celui avec la valeur la plus basse : 2.

On refait pareil après : on parcourt le tableau du graphe en cherchant les arcs partant de 2. On additionne leur valeur avec celle de 2 trouvée juste avant (3 ici). On fait le minimum entre ces valeurs et celle présent la ligne d’au-dessus.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CC | 0 | 1 | 2 | 3 | 4 |
| 0 | 0(0) | 1(0) | 4(0) | 8(0) | 4(0) |
| 0,1 | . | . | 3(1) | 7(1) | 4(0) |
| 0,1,2 | . | . | . | 7(1) | 4(0) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CC | 0 | 1 | 2 | 3 | 4 |
| 0 | 0(0) | 1(0) | 4(0) | 8(0) | 4(0) |
| 0,1 | . | . | 3(1) | 7(1) | 4(0) |
| 0,1,2,4 | . | . | . | 7(1) | . |
| 0,1,2,3,4 | . | . | . | . | . |

On s’arrête quand tous les sommets ont été ajoutés au chemin le plus court OU quand on n’a pas changé de ligne en plusieurs itérations (Exemple : si on a un sommet isolé).

Pour chopper les chemins les plus courts : On parcourt les colonnes du tableau de l’algo :

La première valeur avant un point est la distance, on peut reconstituer les chemin en prenant d’où vient le chemin (le sommet entre parenthèse) et d’aller ensuite d’en la colonne de ce sommet, ainsi de suite jusqu’au sommet initial.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sommets | 0 | 1 | 2 | 3 | 4 |
| Chemins | 0 | 0 | 0,1 | 0,1 | 0 |
| Distances | 0 | 1 | 3 | 7 | 4 |