**RAPPORT DE TP**

**Théorie des graphes**

*TP 3 : Algorithme de Dijkstra*

|  |  |  |
| --- | --- | --- |
| *Auteurs :* |  | *Enseignant :* |
| DUCLOS | Thomas | Mme PALASI |
| KEBE | Ibrahim |  |

**- PARTIE I. Identifier les problématiques, réfléchir, trouver des solutions–**

**- PARTIE II. Recherche des plus courts chemins dans un graphe pondéré :**

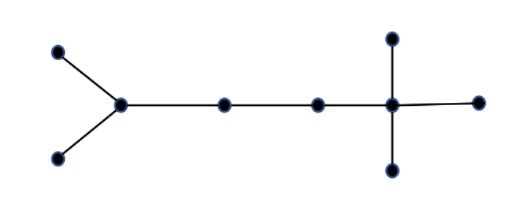
**Algorithme de Dijkstra-**

Nous attestons que ce travail est original, qu’il est le fruit d’un travail commun au binôme et qu’il a été rédigé de manière autonome.

Paris, le 05/03/2022

**PARTIE I. Identifier les problématiques, réfléchir, trouver des solutions**

**Exercice 1 : Robots et chaines de production.**



I

H

G

F

E

D

C

B

A

Afin de définir le meilleur emplacement pour la zone de stockage, il est nécessaire de connaitre le nombre de déplacement d’un sommet a tous les autres pour connaître celui qui le plus optimale sur ce terrain. On fait donc un BFS de notre graphe.

On a pour :

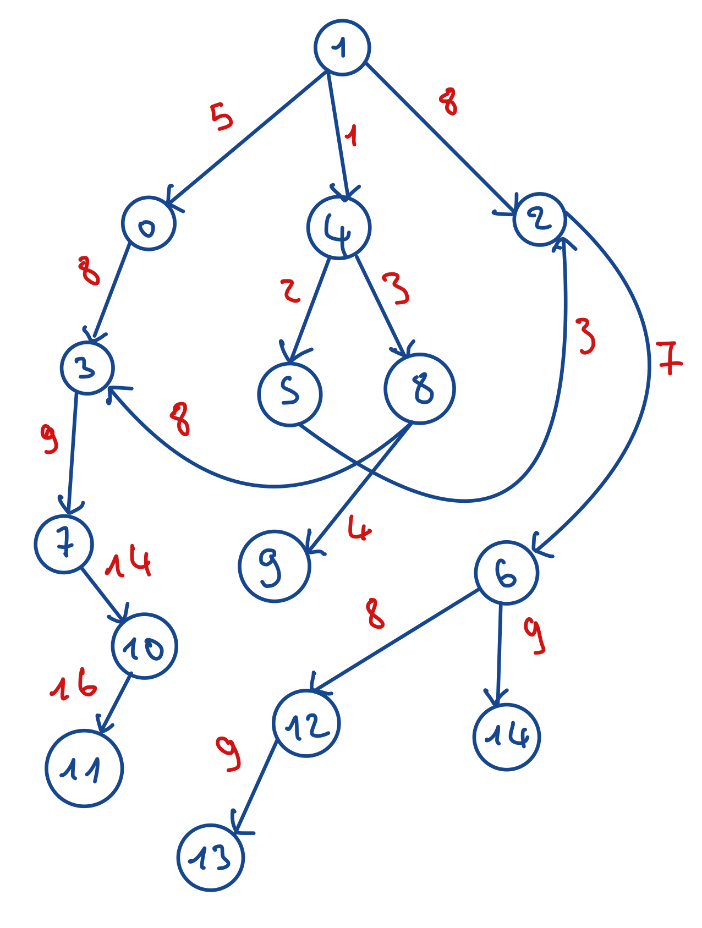
On remarque que le sommet qui accumule le moins de déplacement est le sommet E il est donc le plus optimale.

**Exercice 2 : Comment parcourir un graphe non orienté pondéré, en passant au moins une fois par chaque arête, en revenant à son point de départ, et en minimisant le coût total du parcours ?**

* Nous cherchons ici à passer une seule fois par chaque arête sa implique donc d’avoir un cycle eulérien.
* On peut ici lister 2 situations, la première si 2 sommets sont de degré impair il faut chercher le plus court par cours et le remplacer par un chemin reliant les deux sommets de degré impair, la deuxième situation s’il y a plus de deux sommets de degrés impair il faut lancer l’algorithme de Dijkstra pour retrouver les plus courts chemins entre les degrés impair et les reliés deux à deux en remplaçant les plus courts chemins par des chemins normaux.

**PARTIE II. Recherche des plus courts chemins dans un graphe pondéré : Algorithme de Dijkstra**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOMMET** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** |
| **ETAPE** |  | | | | | | | | | | | | | | |
| **0** | 0 inf ? | 0 0 - | 0 inf ? | 0 inf ? | 0 inf ? | 0 inf ? | 0 inf ? | 0 inf ? | 0 inf ? | 0 inf ? | 0 inf ? | 0 inf ? | 0 inf ? | 0 inf ? | 0 inf ? |
| **1** | 0 5 1 | 1 0 | 0 8 1 |  | 0 1 1 |  |  |  |  |  |  |  |  |  |  |
| **2** |  |  |  |  | 1 1 1 | 0 2 4 |  |  | 0 3 4 |  |  |  |  |  |  |
| **3** |  |  | 0 3 5 |  |  | 1 2 4 |  |  |  |  |  |  |  |  |  |
| **4** |  |  | 1 3 5 |  |  |  | 0 7 2 |  |  |  |  |  |  |  |  |
| **5** |  |  |  | 0 13 8 |  |  |  |  | 1 3 4 | 0 4 8 |  |  |  |  |  |
| **6** |  |  |  |  |  |  |  |  |  | 1 4 8 |  |  |  |  |  |
| **7** | 1 5 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **8** |  |  |  | 0 8 0 |  |  | 1 7 2 |  |  |  |  |  | 0 8 6 |  | 0 9 6 |
| **9** |  |  |  |  |  |  |  |  |  |  |  |  | 1 8 6 | 0 9 12 |  |
| **10** |  |  |  | 1 8 0 |  |  |  | 0 9 3 |  |  |  |  |  |  |  |
| **11** |  |  |  |  |  |  |  | 1 9 3 |  |  | 0 14 7 |  |  |  |  |
| **12** |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 9 12 |  |
| **13** |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 9 6 |
| **14** |  |  |  |  |  |  |  |  |  |  | 1 14 7 | 0 16 10 |  |  |  |
| **15** |  |  |  |  |  |  |  |  |  |  |  | 1 16 10 |  |  |  |

2-

*Arborescence de l’algorithme de Dijkstra*