

**Table des matières**

[**I-Introduction :** 3](#_Toc135268111)

[**1.** **Définition :** 3](#_Toc135268112)

[**2.** **le fonctionnement d'algorithme de Dijkstra :** 3](#_Toc135268113)

[**3.** **Quand utiliser Dijkstra ?** 4](#_Toc135268114)

[**4.** **Complexité :** 4](#_Toc135268115)

[**5.** **L'ALGORITHME DE DIJKSTRA :** 4](#_Toc135268116)

[**II-Exercice d’application** 6](#_Toc135268117)

# **I-Introduction :**

## **Définition :**

L'algorithme de Dijkstra est un algorithme classique utilisé pour trouver le chemin le plus court entre deux nœuds dans un graphe pondéré. Il est largement utilisé dans les problèmes de routage et d'optimisation.

## **le fonctionnement d'algorithme de Dijkstra :**

1-Initialisez un tableau de distances pour tous les nœuds du graphe. La distance initiale pour le nœud de départ est de 0, et toutes les autres distances sont initialement infinies.

2-Marquez tous les nœuds comme non visités.

3-Tant qu'il y a des nœuds non visités :

a. Sélectionnez le nœud non visité avec la plus petite distance, appelez-le "nœud actuel".

b. Marquez le nœud actuel comme visité.

c. Pour chaque nœud voisin non visité du nœud actuel :

* Calculez la distance temporaire en ajoutant la distance du nœud actuel au poids de l'arête qui les relie.
* Si la distance temporaire est inférieure à la distance enregistrée pour ce voisin, mettez à jour la distance enregistrée.

4-Une fois que tous les nœuds ont été visités ou si la distance du nœud cible est infinie, l'algorithme de Dijkstra est terminé.

Après l'exécution de l'algorithme, vous pouvez obtenir le chemin le plus court en remontant à partir du nœud cible vers le nœud de départ en utilisant les informations de distance enregistrées.

## **Quand utiliser Dijkstra ?**

Dijkstra est un algorithme qui est utilisé dans plusieurs situations réelles. Avec les services de cartographie numérique dans Google Maps nous avons la possibilité de trouver la distance d'une ville à une autre ou de notre position de départ jusqu'à l'emplacement désiré le plus proche. Parmi tous les chemins qui relient l'emplacement de départ à celui souhaité, l'application accorde la priorité à celui ayant une distance minimale nommée dist. Elle utilise l'algorithme de Dijkstra pour indiquer l'itinéraire le plus court après avoir visité les sommets ou établit une liste des chemins les plus appropriés à partir de votre noeud de départ.

Les suggestions de liste d'amis sur les réseaux sociaux sont généralement effectuées en utilisant le même principe. L'algorithme de Dijkstra peut être appliqué pour trouver le chemin le plus court entre les utilisateurs, mesuré ici non par la distance, mais par les connexions existantes entre eux. Les drones automatisés utilisés pour une tâche spécifique comme la livraison de colis font également usage de cet algorithme. Une fois la source et la destination connues, l'application de Dijkstra permet à l'appareil de se déplacer dans la direction choisie en suivant le chemin le plus court pour réaliser sa mission en un temps minimal.

## **Complexité :**

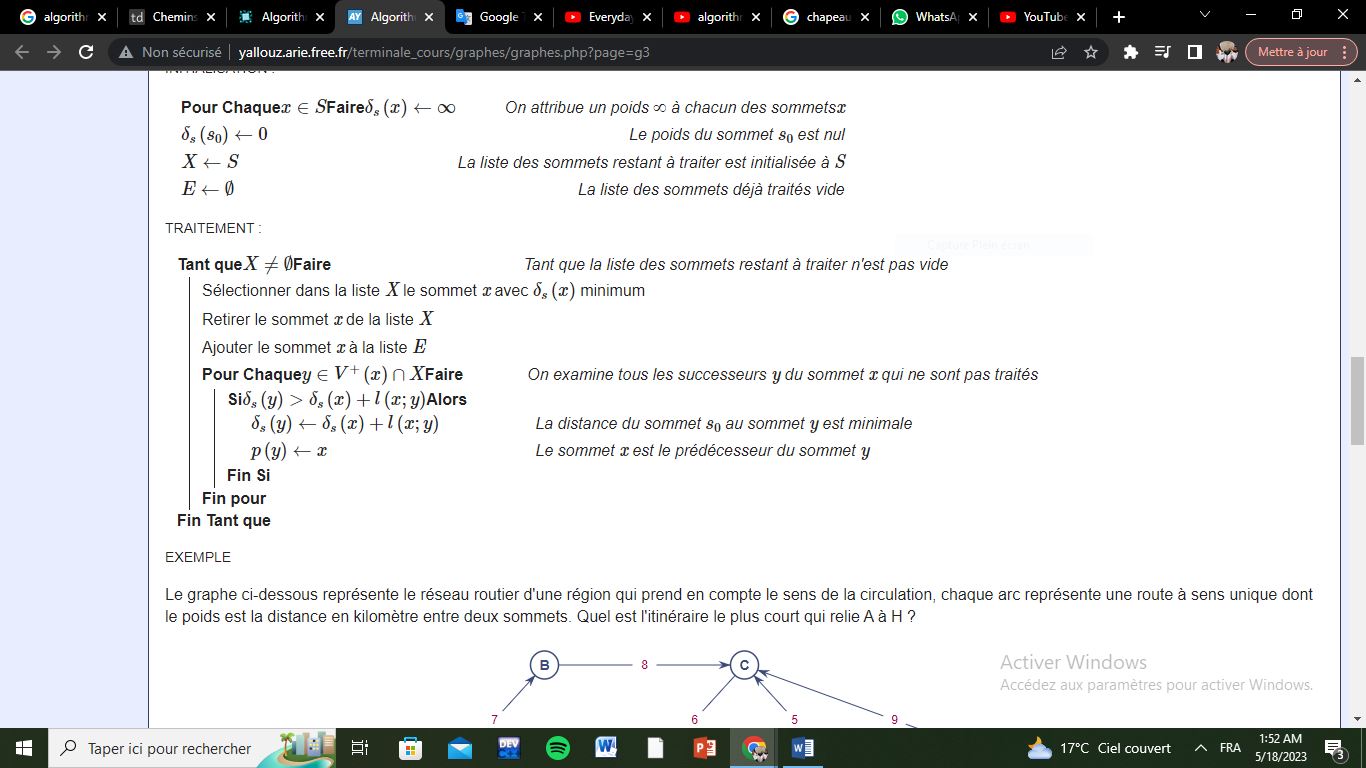
La complexité de l’algorithme dépend fortement de la file de priorité utilisée.On peut montrer qu’on peut atteindre O(|E| + |V|ln(|V|) en pratique. Dans le cadre du cours où le graphe est un labyrinthe et chaque sommet ne peut avoir que 4 voisins au maximum, on obtient donc une complexité O(|V|ln(|V|), car |E| < 4|V|. Cette complexité est moins que quadratique en le nombre de sommets du graphe, ce qui la rend très raisonnable pour notre application.

## **L'ALGORITHME DE DIJKSTRA :**

Soit G un graphe connexe dont les arêtes sont pondérées par des nombres positifs.

NOTATIONS :

* S la liste des sommets du graphe ;
* s0 le sommet du graphe à partir duquel on veut déterminer les plus l(x;y) courts chemins aux autres sommets ;
* δs(x) le poids de l'arête entre deux sommets x et y ;
* V^+(x) la longueur d'un chemin du sommets
* p(x) au sommet x ;
* la liste des successeurs du sommet x ;
* le prédécesseur du sommet x ;
* X liste des sommets restant à traiter ;
* E liste des sommets déjà traités.

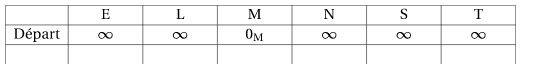
**TRAITEMENT :**

## **INITIALISATION :**

On construit un tableau ayant pour colonnes chacun des sommets du graphe. On ajoute à gauche une colonne qui recensera les sommets choisis à chaque étape (cette colonne est facultative mais facilitera la compréhension de l'algorithme).

Puisque l'on part du sommet M, on inscrit, sur la première ligne intitulée « Départ », 0M0​M​​ **dans la colonne M et \bm{\infty} dans les autres colonnes**.

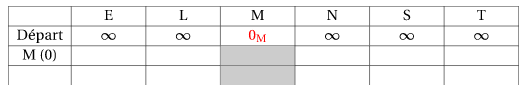
Cela signifie qu'à ce stade, on peut rejoindre M en 0 minute et on n'a rejoint aucun autre sommet puisque l'on n'a pas encore emprunté de chemin...



## **ÉTAPE 1 :**

On sélectionne **le plus petit résultat** de la dernière ligne. Ici, c'est « 0M0​M​​ » qui correspond au chemin menant au **sommet M** en 0 minute.

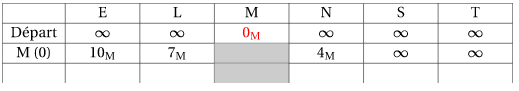
* **On met en évidence cette sélection** (nous l'écrirons en rouge mais il est également possible de la souligner, de l'entourer, etc.).
* **On inscrit le sommet retenu et la durée correspondante dans la première colonne** (ici on écrit M(0)).
* **On désactive les cases situées en dessous de notre sélection** en les grisant par exemple. En effet, on a trouvé le trajet le plus court menant à M ; il sera inutile d'en chercher d'autres.



À partir de M, on voit sur le graphe que l'on peut rejoindre E, L et N en respectivement 10, 7 et 4 minutes. Ces durées sont les durées les plus courtes ; elles sont inférieures au durées inscrites sur la ligne précédente qui étaient « ∞∞ ».

**On inscrit donc \bm{10\_{\text{M}}, 7\_{\text{M}}} et \bm{4\_{\text{M}}} dans les colonnes E, L et N**. Le M situé en indice signifie que l'on vient du sommet M.

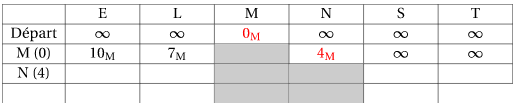
Enfin on complète la ligne en recopiant dans les cellules vides les valeurs de la ligne précédente.



## **ÉTAPE 2 :**

On sélectionne **le plus petit résultat** de la dernière ligne. Ici, c'est « 4M4​M​​ » qui correspond au chemin menant au **sommet N** en 4 minutes.

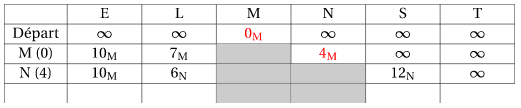
* **On met en évidence cette sélection**.
* **On inscrit le sommet retenu et la durée correspondante dans la première colonne** : N (4).
* **On désactive les cases situées en dessous de notre sélection**. On a trouvé le trajet le plus court menant à N ; il dure 4 minutes.



À partir de N, on peut rejoindre L et S (on ne se préoccupe plus de M qui a été « désactivé »).

* **Si l'on rejoint L :** On mettra 2 minutes pour aller de N à L et 4 minutes pour aller de M à N (ces 4 minutes sont inscrites dans la première colonne) soit au total 6 minutes. **Ce trajet est plus rapide que le précédent** qui durait 7 minutes. **On indique donc \bm{6\_{\text{N}}} dans la colonne L**. Le N situé en indice signifie que l'on vient du sommet N.
* **Si l'on rejoint S :** On mettra 8 minutes pour aller de N à S et 4 minutes pour aller de M à N soit au total 12 minutes. **Ce trajet est plus rapide que le précédent** qui était ∞∞. **On indique donc \bm{12\_{\text{N}}} dans la colonne S**.

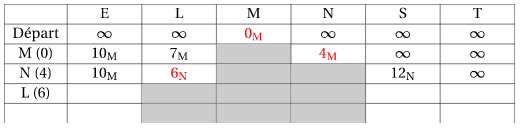
Puis on complète la ligne en recopiant dans les cellules vides les valeurs de la ligne précédente



## **ÉTAPE 3 :**

On sélectionne **le plus petit résultat** de la dernière ligne. Ici, c'est « 6N6​N​​ » qui correspond au chemin menant au **sommet L** en 6 minutes.

* **On met en évidence cette sélection**.
* **On inscrit le sommet retenu et la durée correspondante dans la première colonne** : L (6).
* **On désactive les cases situées en dessous de notre sélection**. On a trouvé le trajet le plus court menant à L ; il dure 6 minutes.



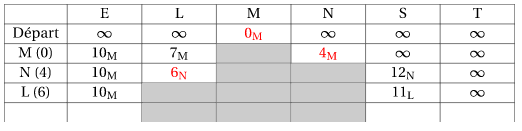
À partir de L, on peut rejoindre E et S (on ne se préoccupe plus de M ni de N qui ont été « désactivés »).

* **Si l'on rejoint E :** On mettra 8 minutes pour aller de L à E et 6 minutes pour aller de M à L soit, au total, 14 minutes. **Ce trajet N'EST PAS plus rapide que le précédent** qui durait 10 minutes.

**On se contente donc de recopier le contenu précédent \bm{10\_{\text{M}}} dans la colonne E**.

* **Si l'on rejoint S :** On mettra 5 minutes pour aller de L à S et 6 minutes pour aller de M à L soit au total 11 minutes. **Ce trajet est plus rapide que le précédent** qui durait 12 minutes. **On indique donc \bm{11\_{\text{L}}} dans la colonne S**.

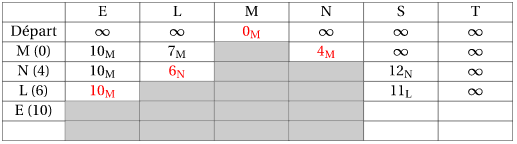
Puis on complète la ligne en recopiant dans les cellules vides les valeurs de la ligne précédente.



## **ÉTAPE 4 :**

On sélectionne **le plus petit résultat**. C'est « 10M10​M​​ » qui correspond au chemin menant au **sommet E** en 10 minutes.

* **On met en évidence cette sélection**.
* **On inscrit le sommet retenu et la durée correspondante dans la première colonne** : E (10).
* **On désactive les cases situées en dessous de notre sélection**. On a trouvé le trajet le plus court menant à E ; il dure 10 minutes.

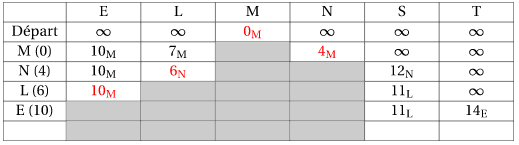


À partir de E, on peut rejoindre S et T (on ne se préoccupe plus des autres sommets qui ont été « désactivés »).

* **Si l'on rejoint S :** On mettra 10 minutes pour aller de E à S et 10 minutes pour aller de M à E (ces 10 minutes sont inscrites dans la première colonne) soit au total 20 minutes.

**Ce trajet N'EST PAS plus rapide que le précédent** qui durait 11 minutes. **On se contente donc de recopier le contenu précédent \bm{11\_{\text{L}}} dans la colonne S**.

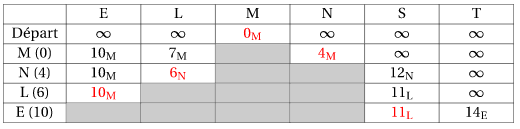
* **Si l'on rejoint T :** On mettra 4 minutes pour aller de E à T et 10 minutes pour aller de M à E soit au total 14 minutes. **Ce trajet est plus rapide que le précédent** qui était ∞∞. **On indique donc \bm{14\_{\text{E}}} dans la colonne T**.



## **ÉTAPE 5 :**

On sélectionne **le plus petit résultat**. C'est « 11L11​L​​ » qui correspond au chemin menant au **sommet S** en 11 minutes.

On a trouvé le trajet le plus court menant à S : il dure **11 minutes**. Comme c'est la question posée dans l'énoncé, il est inutile d'aller plus loin et le tableau est terminé !



Il reste toutefois à reconstituer le trajet qui correspond à cette durée de 11 minutes. En pratique, il est plus facile de trouver le trajet en sens inverse en « remontant » dans le tableau de la façon suivante :

* On part de notre point d'arrivée : **S**
* On recherche la cellule marquée en rouge de la colonne **S** ; elle contient 11L11​L​​. On note la lettre écrite en indice : **L**.
* On recherche la cellule marquée en rouge de la colonne **L** ; elle contient 6N6​N​​. On note la lettre écrite en indice : **N**.
* On recherche la cellule marquée en rouge de la colonne **N** ; elle contient 4M4​M​​. On note la lettre écrite en indice : **M**.

On est arrivé à notre point de départ M après être passé par N et L et S (liste obtenue en listant les sommets en ordre inverse).

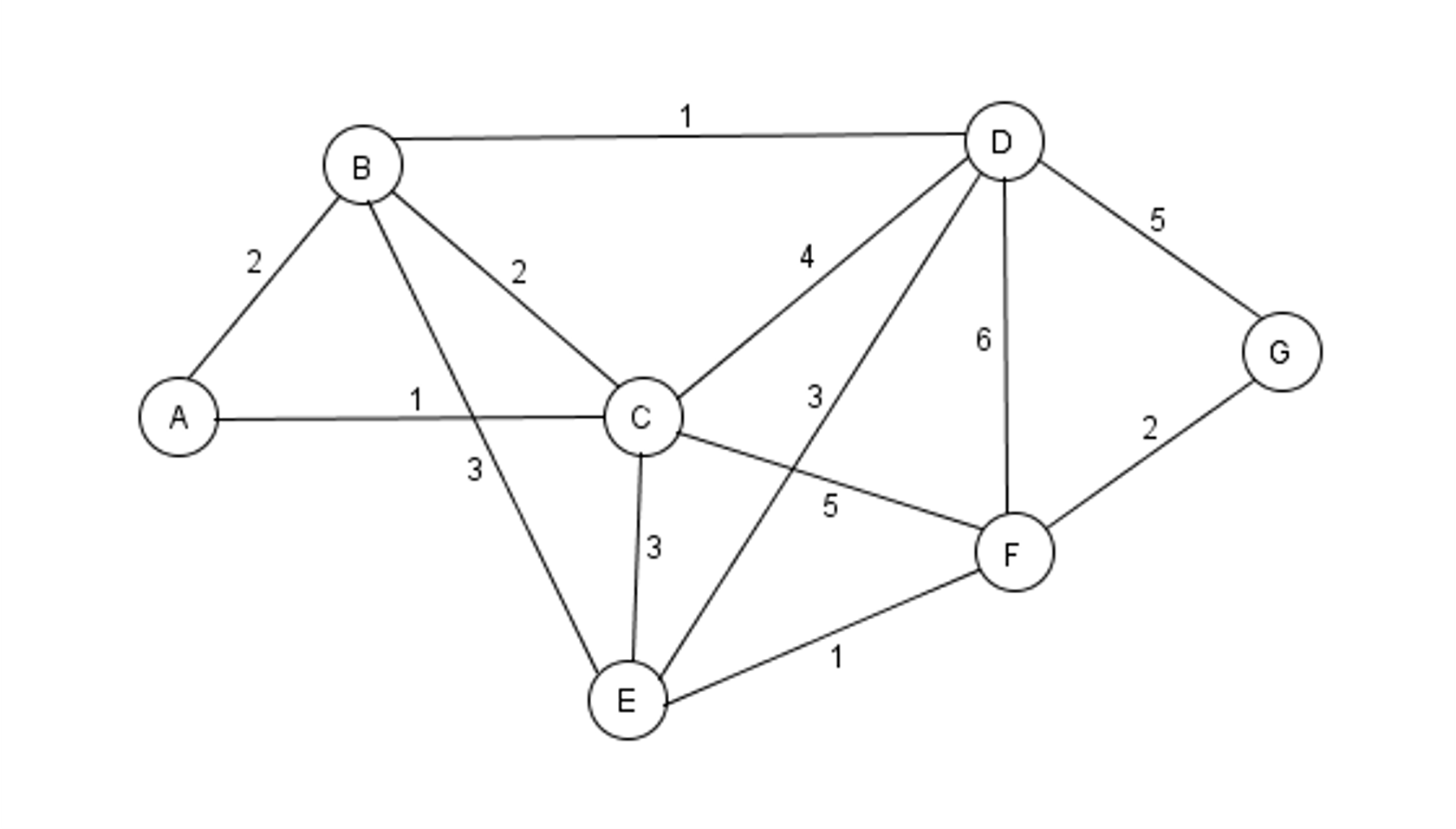
Le trajet optimal est donc **M - N - L - S**.

Enfin, on peut vérifier sur le graphe que ce trajet est correct et dure 11 minutes !

# **II-Exercice d’application**

Le graphe ci-dessous représente le plan d'une ville.

Le sommet A désigne l'emplacement des services techniques.

Les sommets B, C, D, E, F et G désignent les emplacements de jardins publics. Une arête représente l'avenue reliant deux emplacements et est pondérée par le nombre de feux tricolores situés sur le trajet.

**Les parties I et II sont indépendantes.**

**Partie I**

On s'intéresse au graphe non pondéré.

1. Répondre sans justification aux quatre questions suivantes :
   1. Ce graphe est-il connexe ?
   2. Ce graphe est-il complet ?
   3. Ce graphe admet-il une chaîne eulérienne ?
   4. Ce graphe admet-il un cycle eulérien ?
2. Déterminer, en justifiant, le nombre chromatique de ce graphe.

**Partie II**

On s'intéresse au graphe pondéré.

Proposer un trajet comportant un minimum de feux tricolores reliant A à G.

La réponse sera justifiée par un algorithme.

**Corrigé :**

## **PARTIE 1**

* 1. Le graphe est connexe car pour toute paire de sommets, il existe une chaîne reliant ces sommets.
  2. Le graphe n'est pas complet car les points A et D (par exemple) ne sont pas relié par une arête.
  3. D'après le théorème d'Euler, le graphe admet une chaîne eulérienne. En effet, il n'existe que deux sommets de degré impair(C et D)
  4. D'après le théorème d'Euler, le graphe n'admet pas de cycle eulérien. Il faudrait pour cela qu'il n'existe aucun sommet de degré impair

1. BCDE forme un sous-graphe complet. Le nombre chromatique du graphe est donc supérieur ou égal à 4.

On peut colorier le graphe avec 4 couleurs de la façon suivante (par exemple) :

rouge: A, D  
bleu: B, F  
vert: C, G  
jaune: E

Le nombre chromatique du graphe est donc 4.

## **PARTIE 2**

On utilise l'algorithme de Dijkstra :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F | G |
| 0 | 2 (A) | **1 (A)** | ∞∞ | ∞∞ | ∞∞ | ∞∞ |
|  | **2 (A)** |  | 5 (C) | 4 (C) | 6 (C) | ∞∞ |
|  |  |  | **3 (B)** | 4 (C) | 6 (C) | ∞∞ |
|  |  |  |  | **4 (C)** | 6 (C) | 8 (D) |
|  |  |  |  |  | **5 (E)** | 8 (D) |
|  |  |  |  |  |  | **7 (F)** |

Le trajet ACEFG comporte le nombre minimum de 7 feux.