

# TP Graphes 5

Licence Informatique - 2nde année

Année 2023-2024

L'objectif principal de ce TP est d'implanter l'algorithme de dijkstra de calcul de plus court chemin.

## Préliminaires

Récupérez l'archive jointe à cet énoncé, qui contient le dossier TP5 concernant ce TP. Après extraction des données qui s'y trouvent, vous devez disposer :

- d'un sous-dossier `Src`, qui contient la correction du TP4 ;
- d'un sous-dossier `Data` qui contient des graphes qui serviront à tester vos développements.

**Important** Dans ce TP, le format de stockage des matrices d'adjacence va être modifié, afin de pouvoir contenir des graphes de très grande taille. Les fichiers d'exemple fournis sont donc de deux types :

- deux fichiers (`matrice01.txt` et `matrice02.txt`) sont dans le format que vous avez actuellement utilisé et peuvent être utilisés avec la correction fournie ;
- quatre autres fichiers (`graphe01.txt`, `graphe02.txt`, `graphe03.txt` et `graphe04.txt`) sont dans le nouveau format et ne pourront être utilisés qu'après avoir réalisé le premier exercice de cet énoncé. On précise également que les fichiers `graphe01.txt` et `graphe02.txt` correspondent au nouveau format des fichiers `matrice01.txt` et `matrice02.txt`.

## Exercice 1

Le format de fichier d'une matrice d'adjacence, tel qu'il a été utilisé dans les TPs précédents, impose de faire figurer l'ensemble des coefficients de la matrice, même si ceux-ci sont nuls. Cela ne pose pas de problème pour les matrices de petite taille, mais peut rapidement devenir prohibitif pour les matrices de grande taille, souvent creuses. On propose donc, dans cet exercice, de changer la manière dont les matrices d'adjacence sont décrites dans les fichiers texte, en utilisant un nouveau format, qui sera constitué des éléments suivants :

- l'ordre de la matrice (un entier, situé sur la première ligne du fichier) ;
- **pour chaque ligne de la matrice**, la succession des numéros de colonne et des coefficients correspondants. Chaque ligne sera terminée par la valeur -1.

## Exemple

format initial		nouveau format	
7		7	
0 1 0 1 0 0 0		1 1 3 1 -1	<- colonne 1, valeur 1, colonne 3, valeur 1
1 0 0 1 1 1 0		0 1 3 1 4 1 5 1 -1	
0 0 1 0 0 0 0		2 1 -1	
1 1 0 0 1 0 0	=>	0 1 1 1 4 1 -1	
0 1 0 1 0 1 0		1 1 3 1 5 1 -1	
0 1 0 0 1 1 0		1 1 4 1 5 1 -1	
0 0 0 0 0 0 0		-1	<- sommet isolé

L'objectif de cet exercice est donc de modifier la fonction `charger` de manière à pouvoir relire une matrice donnée dans ce nouveau format. Aucune autre partie du code ne doit être modifiée. Vous pourrez tester cette modification sur la correction du TP4 fournie, sur les matrices `graphe01.txt`, `graphe02.txt`, `graphe03.txt` et `graphe04.txt`. **Attention** : la valeur des coefficients peut désormais être différente de 1 ...

## Exemples de résultats

```
./tp4 ../Data/graphe01.txt
0 1 0 1 0 0 0
1 0 0 1 1 1 0
0 0 1 0 0 0 0
1 1 0 0 1 0 0
0 1 0 1 0 1 0
0 1 0 0 1 1 0
0 0 0 0 0 0 0
PARCOURS EN PROFONDEUR RECURSIF
-----
couleurs : N N N N N N N
parents  : X 0 X 1 3 4 X
0
0 1
2
0 1 3
0 1 3 4
0 1 3 4 5
6
PARCOURS EN PROFONDEUR AVEC UNE PILE
-----
couleurs : N N N N N N N
parents  : X 0 X 1 3 4 X
0
0 1
2
0 1 3
0 1 3 4
0 1 3 4 5
6

./tp4 ../Data/graphe02.txt
1 1 0 0 0
1 0 1 0 0
0 0 0 1 0
0 0 0 0 1
0 1 0 0 0
PARCOURS EN PROFONDEUR RECURSIF
-----
couleurs : N N N N N
parents  : X 0 1 2 3
0
0 1
0 1 2
0 1 2 3
0 1 2 3 4
PARCOURS EN PROFONDEUR AVEC UNE PILE
-----
couleurs : N N N N N
parents  : X 0 1 2 3
0
0 1
0 1 2
0 1 2 3
0 1 2 3 4
```

Vous trouverez les résultats à obtenir pour les fichiers `graphe03.txt` et `graphe04.txt` en annexe 1.

## Exercice 2

### Question 1

1. Recopiez dans le sous-dossier `Exercice2` l'ensemble des sources qui se trouvent dans le sous-dossier `Exercice1`, puis renommez le fichier `tp4.cpp` du dossier `Exercice2` en `tp5.cpp` ;
2. Modifiez la fonction `main` de ce second exercice, de telle sorte qu'elle récupère le nom d'un fichier représentant un graphe ainsi qu'un numéro de sommet, charge le graphe correspondant, vérifie que le sommet est compatible avec la taille de la matrice chargée et, si cela est correct, affiche le contenu de la matrice. On précise que le format de lancement de l'application fournie devra être le suivant :

```
./tp5 <graphe> <numéro du sommet de départ>
```

### Question 2

En vous appuyant sur ce qui a été vu en cours sur l'algorithme de Dijkstra, développez les fonctions qui permettent de l'implanter en les ajoutant dans le fichier `dijkstra.cpp` et en modifiant la fonction principale pour appeler la fonction correspondante. Pensez à mettre à jour le fichier `Makefile` en conséquence. **Il est conseillé d'écrire chacune des fonctions nécessaires l'une après l'autre et de les tester avant de passer aux fonctions suivantes.**

Vous pourrez tester votre algorithme sur les fichiers représentant les graphes fournis avec cet énoncé. Les figures 2 et 3 apparaissant en annexe 2 vous donnent un aperçu des deux graphes les plus importants (fichiers `graphe03.txt` et `graphe04.txt`), afin de vérifier les résultats obtenus.

Vous testerez également votre application à partir de différents sommets, celle-ci devant afficher les chemins les plus courts depuis le sommet de départ vers chaque sommet du graphe, dans le format apparaissant dans la figure 1.

```

./tp5 graphe03.txt 0
chemin vers 1 (lg = 85) = 0 1
chemin vers 2 (lg = 217) = 0 2
chemin vers 3 (lg = 503) = 0 2 7 3
chemin vers 4 (lg = 173) = 0 4
chemin vers 5 (lg = 165) = 0 1 5
chemin vers 6 (lg = 403) = 0 2 6
chemin vers 7 (lg = 320) = 0 2 7
chemin vers 8 (lg = 415) = 0 1 5 8
chemin vers 9 (lg = 487) = 0 2 7 9

./tp5 graphe03.txt 7
chemin vers 0 (lg = 320) = 7 2 0
chemin vers 1 (lg = 405) = 7 2 0 1
chemin vers 2 (lg = 103) = 7 2
chemin vers 3 (lg = 183) = 7 3
chemin vers 4 (lg = 493) = 7 2 0 4
chemin vers 5 (lg = 485) = 7 2 0 1 5
chemin vers 6 (lg = 289) = 7 2 6
chemin vers 8 (lg = 251) = 7 9 8
chemin vers 9 (lg = 167) = 7 9

```

FIGURE 1 – Exemples de sortie à obtenir pour le fichier graphe03.txt fourni.

## Annexe1

### Résultat de l'exercice 1 pour le fichier graphe03.txt

```

./tp4 ../Data/graphe03.txt
0 85 217 0 173 0 0 0 0 0
85 0 0 0 0 80 0 0 0 0
217 0 0 0 0 0 186 103 0 0
0 0 0 0 0 0 0 183 0 0
173 0 0 0 0 0 0 0 0 502
0 80 0 0 0 0 0 0 250 0
0 0 186 0 0 0 0 0 0 0
0 0 103 183 0 0 0 0 0 167
0 0 0 0 0 250 0 0 0 84
0 0 0 0 502 0 0 167 84 0
PARCOURS EN PROFONDEUR RECURSIF
-----
couleurs : N N N N N N N N N N
parents : X 0 7 7 9 1 2 9 5 8
0
0 1
0 1 5 8 9 7 2
0 1 5 8 9 7 3
0 1 5 8 9 4
0 1 5
0 1 5 8 9 7 2 6
0 1 5 8 9 7
0 1 5 8
0 1 5 8 9
PARCOURS EN PROFONDEUR AVEC UNE PILE
-----
couleurs : N N N N N N N N N N
parents : X 0 7 7 9 1 2 9 5 8
0
0 1
0 1 5 8 9 7 2
0 1 5 8 9 7 3
0 1 5 8 9 4
0 1 5
0 1 5 8 9 7 2 6
0 1 5 8 9 7
0 1 5 8
0 1 5 8 9

```



## Annexe2

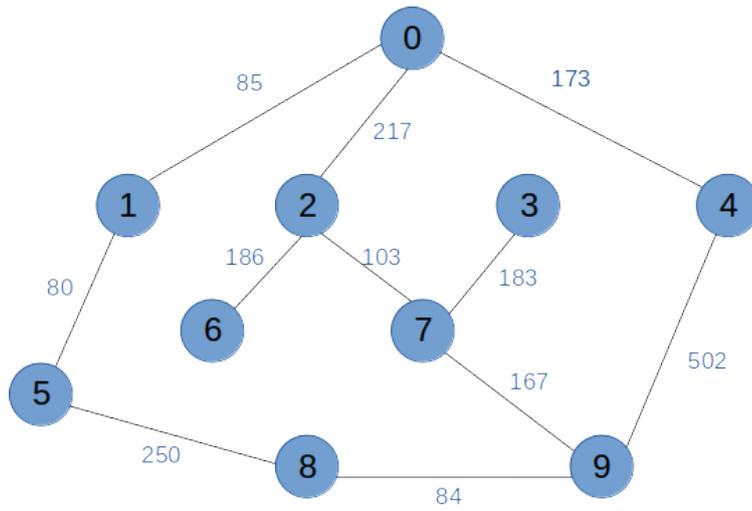


FIGURE 2 – Schéma du graphe fourni dans le fichier `graphe03.txt`.

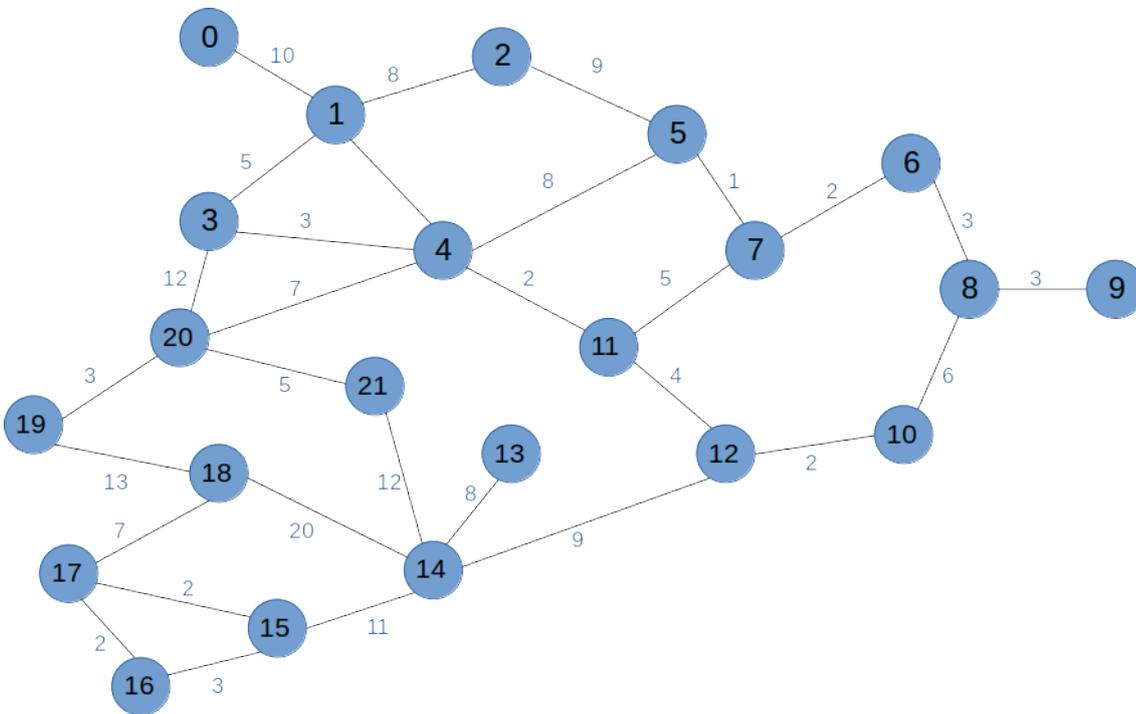


FIGURE 3 – Schéma du graphe fourni dans le fichier `graphe04.txt`.