



02/03/2019

Théorie des Graphes

Examen TP



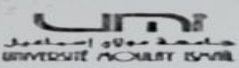
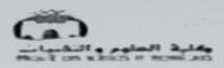
Hassan Zekkouri

RESPONSABLE : PROF. M. D. AOURAGH

| | |
|---|----|
| Examen de théorie des graphes Master SIDI 2019..... | 2 |
| Enoncé : | 2 |
| Exercice 1 : | 4 |
| Exercice 2 : | 8 |
| Exercice 3 : | 9 |
| Exercice 4 : | 10 |

Examen de théorie des graphes Master SIDI 2019

Enoncé :

 
UNIVERSITÉ ALGER
UNIVERSITÉ DE BISKRA

Examen de théorie des graphes Master 1 SIDI –durée 2h le 28 février 2019

Prof. M. D. AOURAGH

Tout document est autorisé

Questions du cours :

1. Préciser si les graphes de la figure 1 sont eulériens, hamiltoniens ou les deux
2. Pour chaque graphe, qu'il est son ordre, le degré de chaque sommet ?

Exercice 1

Pour ce graphe non orienté à 14 sommets de la figure 2 les voisins de chaque sommet sont supposés écrits dans l'ordre croissant de leurs numéros. Ainsi 0 a pour voisins 1, 4, 7, 8 ; 1 a pour voisins 0, 5, 7 ; 2 a pour voisins 5, 10, 12, 13 ; etc.

- 2.1. En partant du sommet 0, faire une exploration en profondeur de ce graphe, en utilisant l'ordre de voisins tel qu'il a été défini. Dessiner l'arbre obtenu.
- 2.2. Toujours en partant du sommet 0, faire une exploration en largeur du graphe. On aura intérêt à utiliser l'évolution d'une file, afin de dessiner l'arbre final de l'exploration.

Exercice 2

On veut construire un réseau (figure 3) avec un coût minimum pour relier 12 commutateurs. Les coûts de câblage sont donnés par le graphe G. Suite à une décision politique les liaisons câblées GH et AE sont imposées. Déterminer alors un câblage à coût minimal respectant ces contraintes. (Vous préciserez l'algorithme utilisé, l'adaptation de cet algorithme au cas précis de l'exercice et enfin son application étape par étape.)

Exercice 3

Le graphe de la figure 4 représente des temps de vol (en heures) de liaisons aériennes entre 8 aéroports.

1. Déterminer les trajets les plus rapides depuis A vers chacune des 7 autres villes. (Vous préciserez l'algorithme utilisé et les étapes de son application).
2. On impose maintenant un temps d'escale dans chaque aéroport selon le tableau suivant

| aéroport | B | C | D | E | F | G | H |
|----------------|----|----|----|----|----|----|----|
| temps d'escale | 2h | 2h | 3h | 1h | 2h | 5h | 2h |

Que doit-on modifier sur le graphe ? Quel est le trajet le plus rapide depuis A vers H ?

Exercice 4

Sept agences de voyage proposent des visites de monuments et lieux emblématiques de Saint-Petersbourg : la cathédrale Saint-Isaac, le Musée de l'Ermitage, le Musée russe et la forteresse Pierre et Paul. Un même lieu ne peut pas être visité par plusieurs groupes de compagnies différentes le même jour. La première compagnie fait visiter uniquement la cathédrale Saint-Isaac ; la seconde la cathédrale Saint-Isaac et le Musée russe ; la troisième la forteresse Pierre et Paul ; la quatrième le Musée russe et la forteresse Pierre et Paul ; la cinquième la cathédrale Saint-Isaac et le Musée de l'Ermitage ; la sixième le Musée de l'Ermitage et la forteresse Pierre et Paul ; la septième le Musée russe et le Musée de l'Ermitage. (Voir le modèle de la figure 5).

Ces agences peuvent-elles organiser les visites sur les trois premiers jours de la semaine ?

Figures : Examen de la théorie des graphes- Master 1 SIDI février 2019

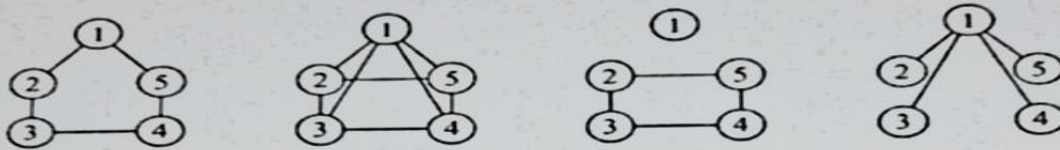


Figure 1

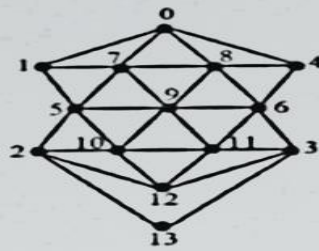


Figure 2

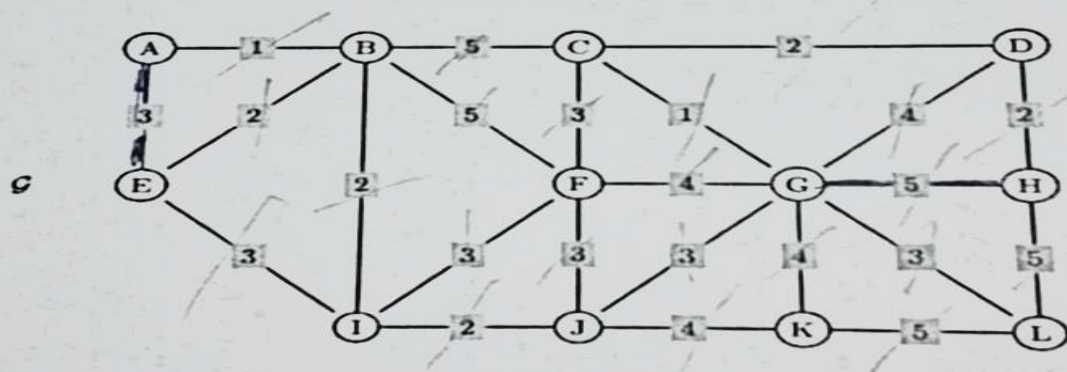


Figure 3

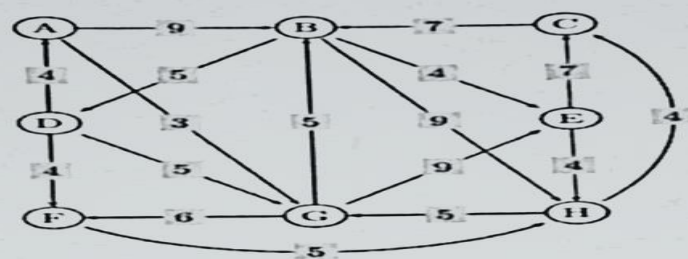


Figure 4

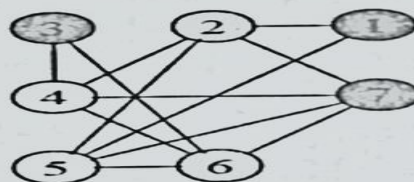


Figure 5

Exercice 1 :

1) Parcours en profondeur à partir du sommet 0 :

a) Première algorithm (Pile) :

Utilisation d'une pile : le dernier sommet pas encore ajouter au pile et qui est adjacent au sommet dernièrement afficher sera à la tête de la pile est donc le suivant à être affiché. C'est toujours le sommet de numérotation le plus grand non encore ajouté à la pile !

On affiche les sommets en ordre d'exploration on aura la suite suivante :

0-8-9-11-12-2-13-3-10-5-6-7-4-1

```

"D:\0. MST SIDI\1. Théorie de Graphe\TP\C\Graph-Theory\Exam TP\bin\Debu...
l'arete 13 ---- 12 existe taper son poids sinon taper 0 : Si l'arete 13 ----
13 existe taper son poids sinon taper 0 : Si l'arete 14 ---- 1 existe taper s
on poids sinon taper 0 : Si l'arete 14 ---- 2 existe taper son poids sinon tap
er 0 : Si l'arete 14 ---- 3 existe taper son poids sinon taper 0 : Si l'arete
14 ---- 4 existe taper son poids sinon taper 0 : Si l'arete 14 ---- 5 existe
taper son poids sinon taper 0 : Si l'arete 14 ---- 6 existe taper son poids s
inon taper 0 : Si l'arete 14 ---- 7 existe taper son poids sinon taper 0 : Si
l'arete 14 ---- 8 existe taper son poids sinon taper 0 : Si l'arete 14 ---- 9
existe taper son poids sinon taper 0 : Si l'arete 14 ---- 10 existe taper so
n poids sinon taper 0 : Si l'arete 14 ---- 11 existe taper son poids sinon tap
er 0 : Si l'arete 14 ---- 12 existe taper son poids sinon taper 0 : Si l'arete
14 ---- 13 existe taper son poids sinon taper 0 : Si l'arete 14 ---- 14 exi
ste taper son poids sinon taper 0 :
##### Debut de parcours #####

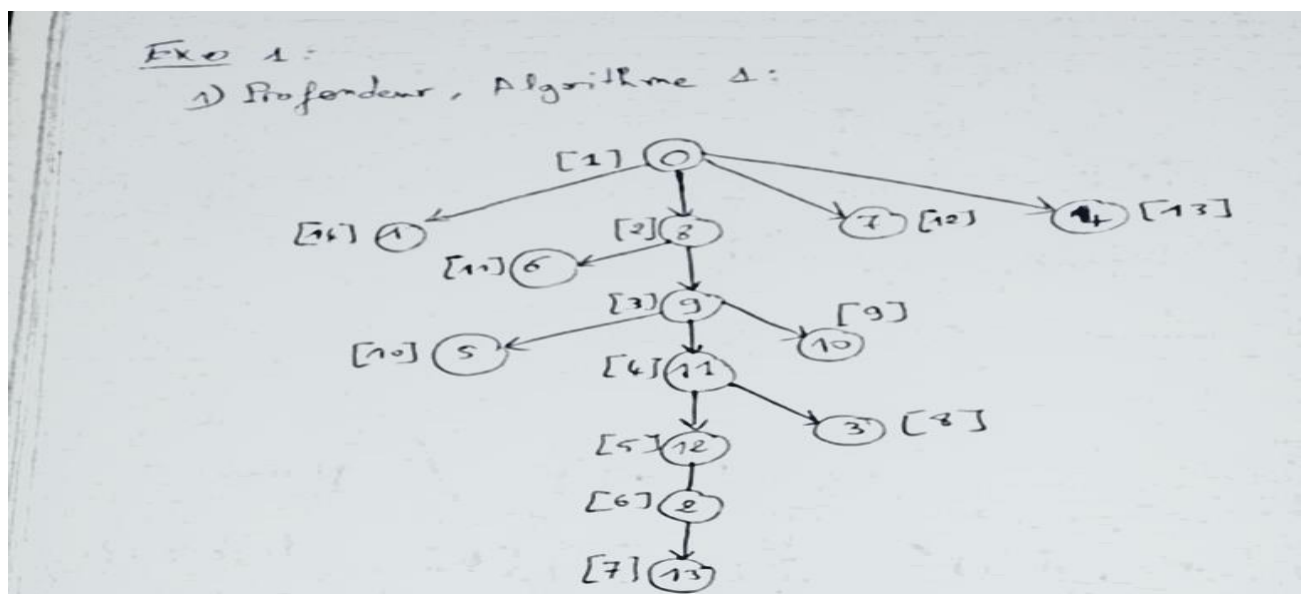
Test de parcoursProfondeur(mgno, i);

Parcours Pour La racine : 0
parcoursProfondeur(mgno, i);
0 8 9 11 12 2 13 3 10 5 6 7 4 1
Fin de parcours en profondeur (DFS)!

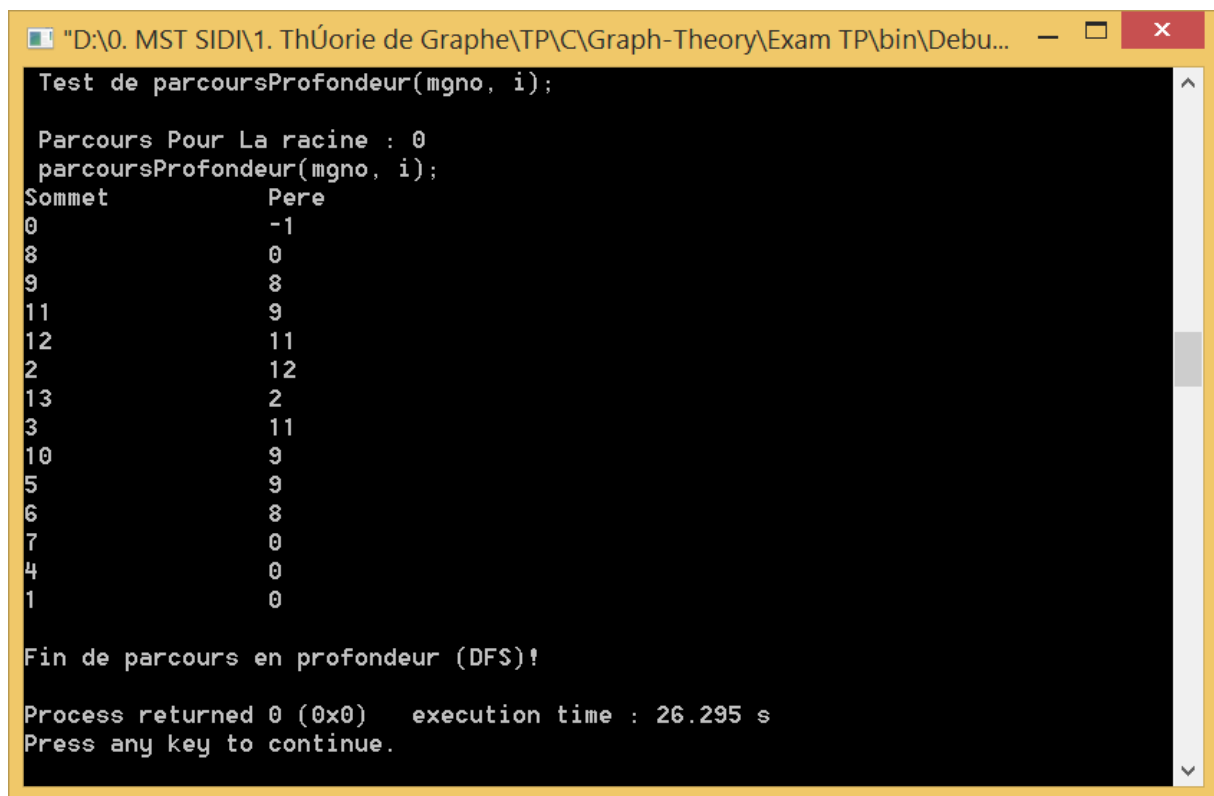
Process returned 0 (0x0)   execution time : 37.551 s
Press any key to continue.

```

En affichant les pères : On aura l'arborescence suivante : [ordre d'affichage]



L'exécution :



```
"D:\0. MST SIDI\1. Théorie de Graphe\TP\C\Graph-Theory\Exam TP\bin\Debu...
Test de parcoursProfondeur(mgno, i);

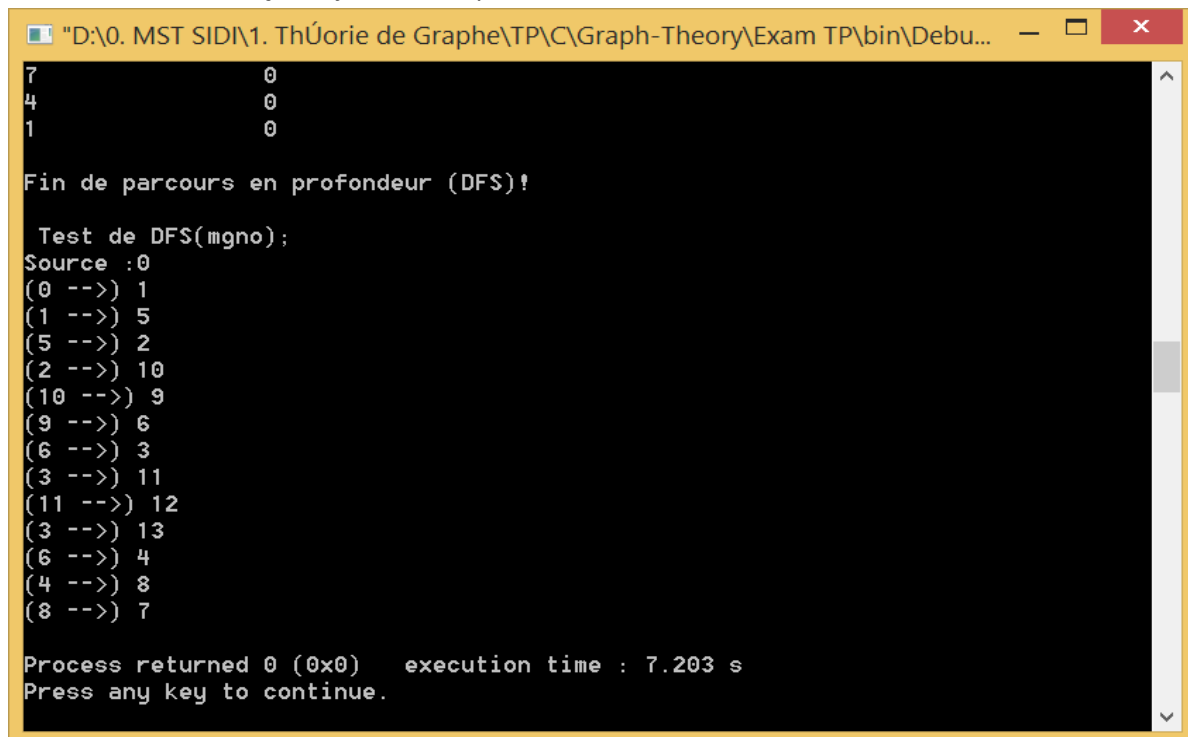
Parcours Pour La racine : 0
parcoursProfondeur(mgno, i);
Sommet      Pere
0            -1
8            0
9            8
11           9
12           11
2            12
13           2
3            11
10           9
5            9
6            8
7            0
4            0
1            0

Fin de parcours en profondeur (DFS)!

Process returned 0 (0x0)   execution time : 26.295 s
Press any key to continue.
```

b) Deuxième algorithme : DFS

Toujours on choisit le plus petit sommet (numéro) adjacent au sommet dernièrement visité, si tous les successeurs sont déjà visités en revient en arrière vers le premier sommet avec des successeurs non encore visités et ça toujours en respectant l'ordre.



```
"D:\0. MST SIDI\1. Théorie de Graphe\TP\C\Graph-Theory\Exam TP\bin\Debu...
7            0
4            0
1            0

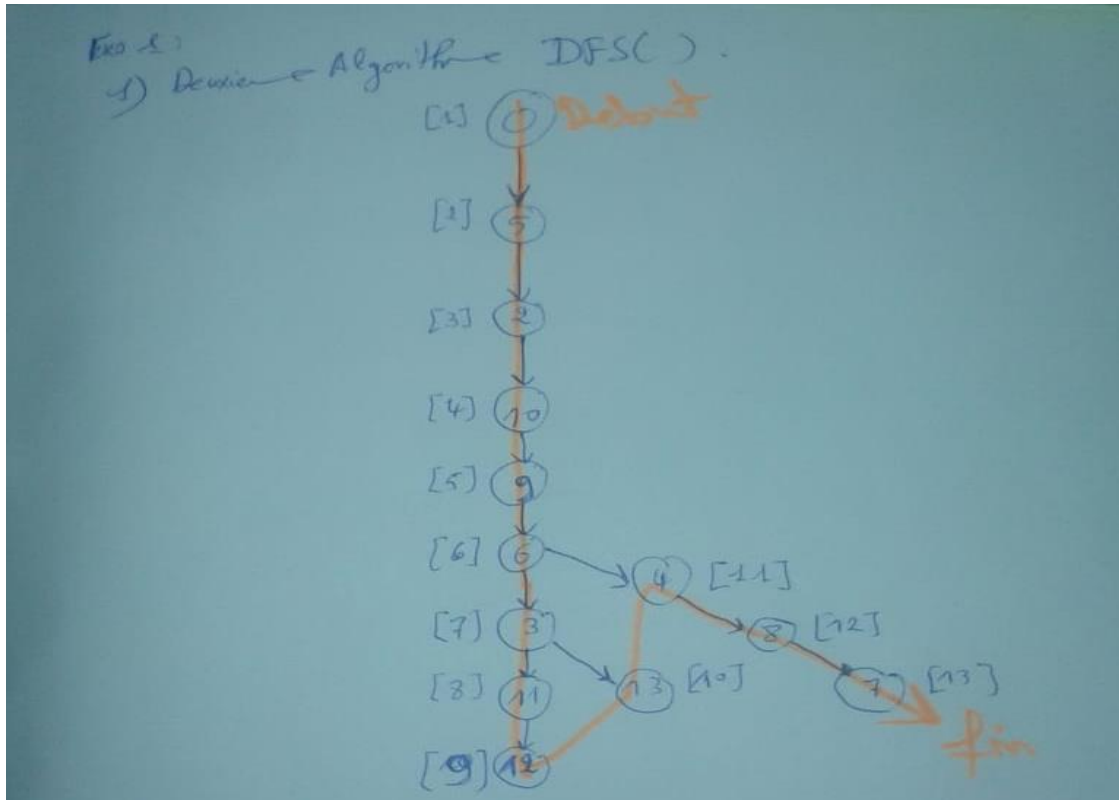
Fin de parcours en profondeur (DFS)!

Test de DFS(mgno);
Source :0
(0 -->) 1
(1 -->) 5
(5 -->) 2
(2 -->) 10
(10 -->) 9
(9 -->) 6
(6 -->) 3
(3 -->) 11
(11 -->) 12
(3 -->) 13
(6 -->) 4
(4 -->) 8
(8 -->) 7

Process returned 0 (0x0)   execution time : 7.203 s
Press any key to continue.
```

On a un autre résultat car le parcours en profondeur dépend de premier successeur choisit.

L'arborescence correspondante :



2) Parcours en largeur à partir du sommet 0 :

Si on affiche les sommets en ordre d'exploration on aura la suite suivante :

0 -> 1 -> 4 -> 7 -> 8 -> 5 -> 6 -> 9 -> 2 -> 10 -> 3 -> 11 -> 12 -> 13

```

"D:\0. MST SIDI\1. Théorie de Graphe\TP\C\Graph-Theory\Exam TP\bin\Debu...
non taper 0 : Si l'arete 13 ---- 11 existe taper son poids sinon taper 0 : Si
l'arete 13 ---- 12 existe taper son poids sinon taper 0 : Si l'arete 13 ----
13 existe taper son poids sinon taper 0 : Si l'arete 14 ---- 1 existe taper s
on poids sinon taper 0 : Si l'arete 14 ---- 2 existe taper son poids sinon tap
er 0 : Si l'arete 14 ---- 3 existe taper son poids sinon taper 0 : Si l'arete
14 ---- 4 existe taper son poids sinon taper 0 : Si l'arete 14 ---- 5 existe
taper son poids sinon taper 0 : Si l'arete 14 ---- 6 existe taper son poids s
inon taper 0 : Si l'arete 14 ---- 7 existe taper son poids sinon taper 0 : Si
l'arete 14 ---- 8 existe taper son poids sinon taper 0 : Si l'arete 14 ---- 9
existe taper son poids sinon taper 0 : Si l'arete 14 ---- 10 existe taper son
poids sinon taper 0 : Si l'arete 14 ---- 11 existe taper son poids sinon tap
er 0 : Si l'arete 14 ---- 12 existe taper son poids sinon taper 0 : Si l'arete
14 ---- 13 existe taper son poids sinon taper 0 : Si l'arete 14 ---- 14 exi
ste taper son poids sinon taper 0 :
##### Debut de parcours #####

Test de parcoursLargeur(mgno, i);

Parcours Pour La racine : 0
0 1 4 7 8 5 6 9 2 10 3 11 12 13
Fin de parcoursLargeur (BFS)!

Process returned 0 (0x0) execution time : 8.959 s
Press any key to continue.
  
```


(tous les codes sont dans le projet codeblocks)

En effectuant un autre parcours où on affiche les pères, on peut tracer l'arborescence correspondant et on ajoute un temps entre crochet pour indiquer l'ordre d'exploration :

```
"D:\0. MST SIDI\1. Théorie de Graphe\TP\C\Graph-Theory\Exam TP\bin\Debu...
##### Debut de parcours #####

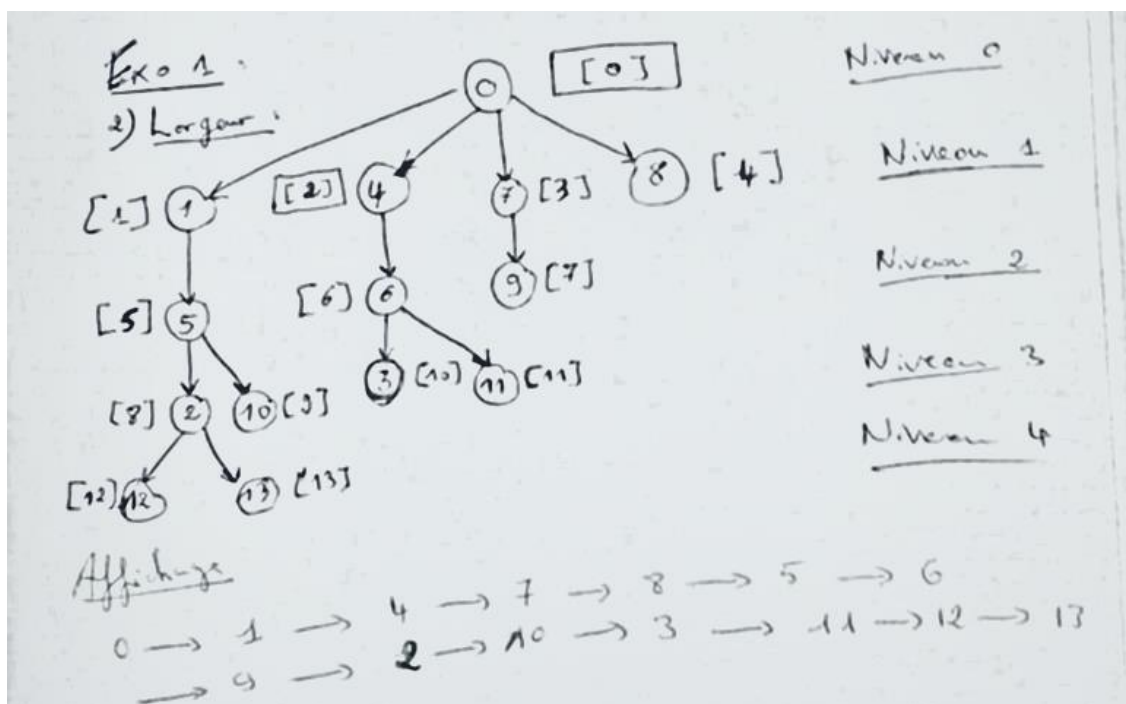
Test de parcoursLargeur(mgno, i);

Parcours Pour La racine : 0
Racine: 0
S(1) : le pere est S(0)
S(4) : le pere est S(0)
S(7) : le pere est S(0)
S(8) : le pere est S(0)
S(5) : le pere est S(1)
S(6) : le pere est S(4)
S(9) : le pere est S(7)
S(2) : le pere est S(5)
S(10) : le pere est S(5)
S(3) : le pere est S(6)
S(11) : le pere est S(6)
S(12) : le pere est S(2)
S(13) : le pere est S(2)

Fin de parcoursLargeur (BFS)!

Process returned 0 (0x0)   execution time : 11.260 s
Press any key to continue.
```

On obtient l'arborescence suivante :



Exercice 2 :

Pour adapter l'algorithme de Kruskal, pour la recherche de l'arbre de poids minimal, à notre problème, on ajoute les deux arêtes imposées en premier !

A—3—E et G—5—H

```
EXAM

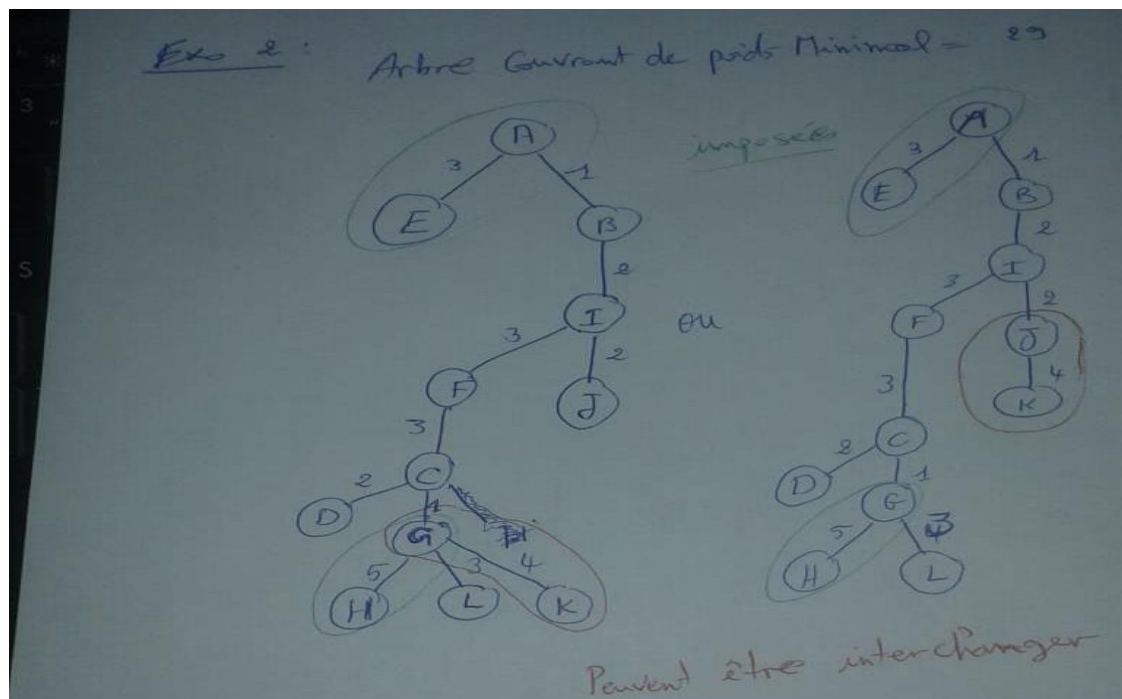
Voici les arcs (aretes) de l'arbre de poids minimal:
A --(3)-- E
G --(5)-- H
G --(3)-- L
A --(1)-- B
C --(1)-- G
B --(2)-- I
I --(2)-- J
C --(2)-- D
I --(3)-- F
C --(3)-- F
K --(4)-- J

Process returned 0 (0x0)   execution time : 0.326 s
Press any key to continue.
```

Voici l'arborescence correspondante :

Remarque !

A la main il est probable d'ajouter l'arête G—4—K à la place de J—4—K, ou encore d'autre sommets de poids égaux !



Exercice 3 :

- 1) Le graphe est à valuations (poids) positives donc on utilise l'algorithme de Dijkstra :

Code → voir le projet Exam TP

Racine : le sommet A

Matrice d'adjacence du graphe :

```

0 9 0 0 0 0 3 0
0 0 0 5 4 0 0 9
0 7 0 0 0 0 0 0
4 0 0 0 0 4 5 0
0 0 7 0 0 0 0 4
0 0 0 0 0 0 0 5
0 5 0 0 9 6 0 0
0 0 4 0 0 0 5 4

```

Résultat d'exécution :

```

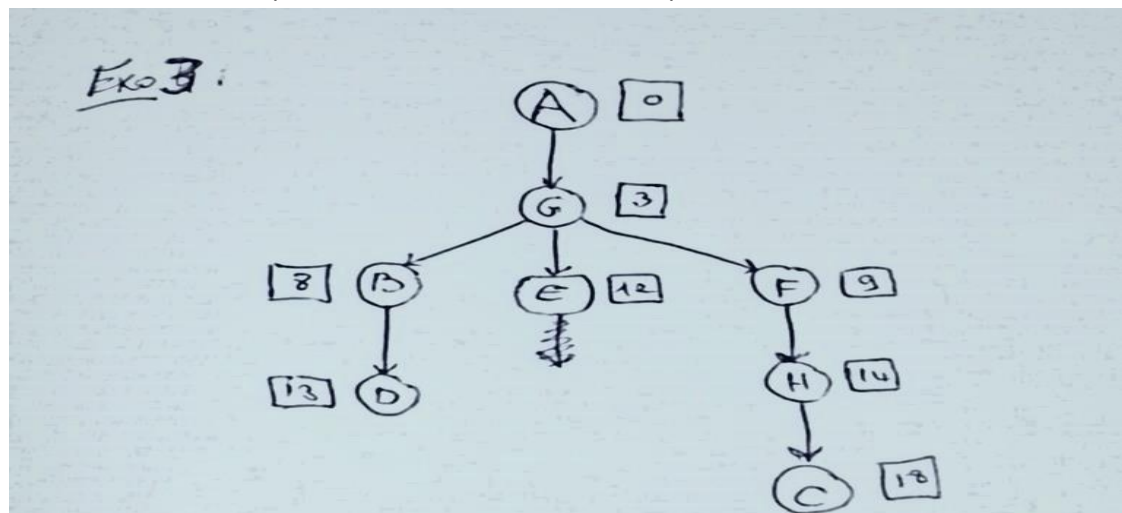
"D:\0. MST SIDI\1. Théorie de Graphe\TP\C\Graph-Theory\Exam TP\bin\Debu...
son poids sinon 0: Si l'arc 7 --> 1 existe taper son poids sinon 0: Si l'arc
7 --> 2 existe taper son poids sinon 0: Si l'arc 7 --> 3 existe taper son poi
ds sinon 0: Si l'arc 7 --> 4 existe taper son poids sinon 0: Si l'arc 7 --> 5
existe taper son poids sinon 0: Si l'arc 7 --> 6 existe taper son poids sino
n 0: Si l'arc 7 --> 7 existe taper son poids sinon 0: Si l'arc 7 --> 8 exist
e taper son poids sinon 0: Si l'arc 8 --> 1 existe taper son poids sinon 0: Si
l'arc 8 --> 2 existe taper son poids sinon 0: Si l'arc 8 --> 3 existe taper
son poids sinon 0: Si l'arc 8 --> 4 existe taper son poids sinon 0: Si l'arc
8 --> 5 existe taper son poids sinon 0: Si l'arc 8 --> 6 existe taper son po
ids sinon 0: Si l'arc 8 --> 7 existe taper son poids sinon 0: Si l'arc 8 -->
8 existe taper son poids sinon 0: Distances a partir de source:

Sommet      Distance      Pere
A           0           null
B           8           G
C           18          H
D           13          B
E           12          G
F           9           G
G           3           A
H           14          F

Process returned 0 (0x0)   execution time : 8.513 s
Press any key to continue.

```

Avec ses Résultats on peut tracer l'arborescence correspondante.



2) Trajet le plus rapide depuis A vers H :

On effectue une mise à jour au niveau des distances :

```

-
0 9 0 0 0 0 3 0
0 0 0 7 6 0 0 11
0 9 0 0 0 0 0 0
7 0 0 0 0 7 8 0
0 0 8 0 0 0 0 5
0 0 0 0 0 0 0 7
0 10 0 0 14 11 0 0
0 0 6 0 0 0 7 6

```

Résultats :

```

son poids sinon 0: Si l'arc 7 --> 1 existe taper son poids sinon 0: Si l'arc
7 --> 2 existe taper son poids sinon 0: Si l'arc 7 --> 3 existe taper son poi
ds sinon 0: Si l'arc 7 --> 4 existe taper son poids sinon 0: Si l'arc 7 --> 5
existe taper son poids sinon 0: Si l'arc 7 --> 6 existe taper son poids sino
n 0: Si l'arc 7 --> 7 existe taper son poids sinon 0: Si l'arc 7 --> 8 exist
e taper son poids sinon 0: Si l'arc 8 --> 1 existe taper son poids sinon 0: Si
l'arc 8 --> 2 existe taper son poids sinon 0: Si l'arc 8 --> 3 existe taper
son poids sinon 0: Si l'arc 8 --> 4 existe taper son poids sinon 0: Si l'arc
8 --> 5 existe taper son poids sinon 0: Si l'arc 8 --> 6 existe taper son po
ids sinon 0: Si l'arc 8 --> 7 existe taper son poids sinon 0: Si l'arc 8 -->
8 existe taper son poids sinon 0: Distances a partir de source:

Sommet      Distance      Pere
A            0            null
B            9            A  Arc 1
C           23            E
D           16            B
E           15            B
F           14            G
G            3            A
H           20            B  Arc 2

Process returned 0 (0x0)   execution time : 6.245 s
Press any key to continue.

```

Effectivement le plus court chemin de A vers H est de coût = 20, ABH.

Exercice 4 :

On a sept agences de voyage et quatre lieux :

| Lieu | Agences qui le visitent |
|---------------------------|-------------------------|
| Cathédrale Saint-Isaac | 1, 2, 5 |
| Musée de l'hermitage | 5, 6, 7 |
| Musée russe | 2, 4, 7 |
| Forteresse Pierre et Paul | 3, 4, 6 |

On considère que les sommets sont les sept agences.

Un lien entre deux sommets x et y existe si au moins un lieu est visité par x et y.

On a alors les résultats sur la **figure 5**.

Partie inférieure de la matrice D'adjacence : (graphe non orienté)

```
0
1 0
0 0 0
0 1 1 0
1 1 0 0 0
0 0 1 1 1 0
0 1 0 1 1 1 0
```

Maintenant, on considère le tableau suivant des degrés des sommets trié en ordre décroissant :

| Sommet | 2 | 4 | 5 | 6 | 7 | 1 | 3 |
|--------|---|---|---|---|---|---|---|
| Degré | 4 | 4 | 4 | 4 | 4 | 2 | 2 |

On va appliquer l'algorithme de Powell-Welsh pour colorier le graphe. Ce qui donne trois couleurs possibles comme sur la figure, alors le graphe peut être partitionné en trois stable ce qui va permettre aux agences d'organiser les visites sur les trois jours de la semaine en respectant la contrainte qui dit « Un même lieu ne peut pas être visité par plusieurs groupes de compagnies différentes le même jour ».

Par exemple :

Agences 1, 3 et 7 le Lundi ;

Agences 2 et 6 le Mardi ;

Et enfin les agences 4 et 5 le Mercredi.

Et voici, le résultat d'exécution :

```
"D:\0. MST SIDI\1. Théorie de Graphe\TP\C\Graph-Theory\Mini Projet 3 - autr...
non taper 0 : Si l'arete 6 ---- 4 existe taper son poids sinon taper 0 : Si l'
arete 6 ---- 5 existe taper son poids sinon taper 0 : Si l'arete 6 ---- 6 ex
iste taper son poids sinon taper 0 : Si l'arete 7 ---- 1 existe taper son poid
s sinon taper 0 : Si l'arete 7 ---- 2 existe taper son poids sinon taper 0 : S
i l'arete 7 ---- 3 existe taper son poids sinon taper 0 : Si l'arete 7 ---- 4
existe taper son poids sinon taper 0 : Si l'arete 7 ---- 5 existe taper son
poids sinon taper 0 : Si l'arete 7 ---- 6 existe taper son poids sinon taper 0
: Si l'arete 7 ---- 7 existe taper son poids sinon taper 0 :
##### Debut de Coloriage #####

Sommet S(1) --> couleur 1
Sommet S(3) --> couleur 1
Sommet S(7) --> couleur 1

Sommet S(2) --> couleur 2
Sommet S(6) --> couleur 2

Sommet S(4) --> couleur 3
Sommet S(5) --> couleur 3

##### FIN de COLORIAGE #####

Process returned 0 (0x0)   execution time : 3.699 s
Press any key to continue.
```

Merci Monsieur pour la partie pratique, c'est mieux qu'à faire le travail à la main!

Fin