Rapport d'Algorithmique Avancée

Encadré par: M.Ouzia et M.Marchetti

Réalisé par :Yining BAO

Sommaire

1.Le problème à traiter

Soit dans un graphe donné G={V,E,ω,c} V est l'ensemble des sommets(1,2,3,4,5,6,7) et E est l'ensemble des arrêts entre les sommets. ω est le poids pour chaque arrêt et c est une application a chaque sommet du graphe associé un coût cj .

Notre but est de former tous les arbres minimaux connectant tous les sommets du sous-ensemble de sommets données T.

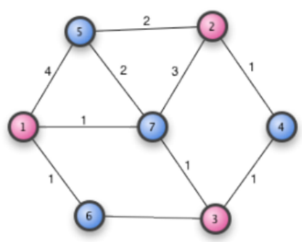


Figure 1: Le Graphe G(par le sujet)

On considere la mesure ∑ωe (e∈E) et l'arbre A={V[A], E[A]} , V[A] est l'ensemble des sommets et E[A] est l'ensemble des arrêts.

On doit trouver les arbres minimaux d'un graphe.

Ce genre de problème peut consister à trouver la meilleure planification du chemin pour le voyage, le réseau , l'électricité, le gaz etc.

2.Analyse du travail à faire

Comme le graphe est un graphe non orienté et trouver l'arbre minimal.

On pense a les deux algorithmes : Kruskal et Prim.

En comparant les nombres des arrêts et des sommets, on a 7 sommets et 10 arrêts. Le nombre de sommets est moins élevé que le nombre d'arrêts, donc on choisit l'algorithme de Prim qui consiste à faire croître un arbre depuis un sommet (Kruskal est à partir des arrêts).

A part un sous-ensemble T qui contient un sommet de départ. A chaque itération, on ajoute le sommet avec le poids minimal à côté du T par les arrêts connectées.

Répéter de cette manière, jusqu'à atteindre tous les sommets et on peut trouver l'arbre minimal.

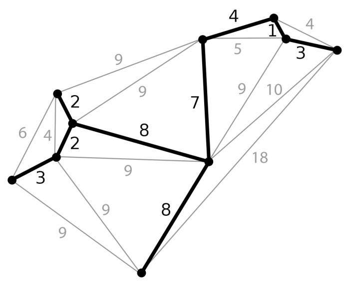


Figure 2: L'algorithme de Prim par étape([Algorithme de Prim — Wikipédia (wikipedia.org)](wikipedia.org))

Pour énumérer toutes les possibilités de l'arbre, on doit utiliser l'algorithme DFS à l'aide de l'arbre couvrant minimum avec le poids minimal pour trouver les arbres avec le poids minimal.

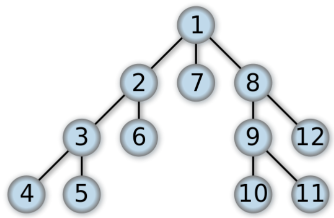


Figure 3: L'algorithme DFS([Depth-first search - Wikipedia](https://en.wikipedia.org/wiki/Depth-first_search))

A partir du graphe G et de l'algorithme de Prim, on peut analyser le graphe.
On choisit le sommet 1 pour démarrer, et l'arrêt entre le sommet 3 et 6 est manqué, on choisit 2 pour cet arrêt

Iteration 1:
Le sommet 1 relie le sommet 5,7 et 6 avec le poids 4,1,1
On choisit soit le sommet 3 soit le sommet 7 avec le poids plus petit 1
Et on choisit 7
Coût de min{4,1,1},Ajoute de {1,7}

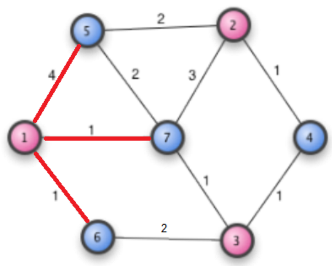
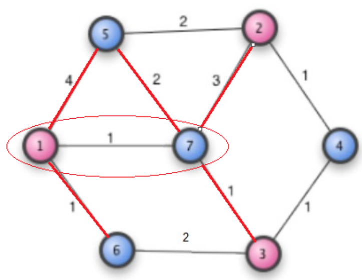


Figure 4: L'itération 1

Iteration 2:
Les sommet 1 et 7 relient le sommet 2,3,5,6 et avec le poids 3,1,2,4,1
On choisit soit le sommet 3 soit le sommet 6 avec le poids plus petit 1
Et on choisit 3,Coût de min{3,1,2,4}. Ajoute de {1,7,3}

Figure 5 L'itération 2



Iteration 3:
Les sommet 1,3 et 7 relient le sommet 2,4,5,6 et avec le poids 3,1,2,2,1
On choisit soit le sommet 4 soit le sommet 6 avec le poids plus petit 1
Et on choisit 4,Coût de main{1,2,3}. Ajoute de {1,7,3,4}

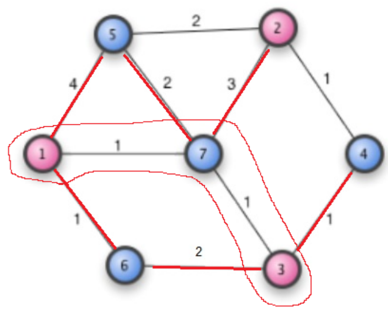


Figure 6: L'itération 3

Iteration 4:

Les sommet 1,3,4 et 7 relient le sommet 2,5,6 et avec le poids 1,3,2,4,1,2

On choisit soit le sommet 2 soit le sommet 6 avec le poids plus petit 1

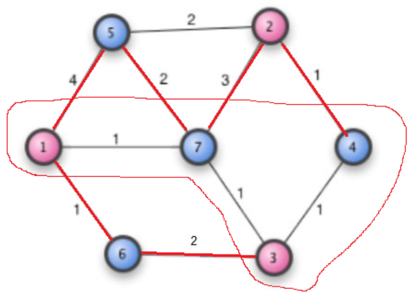


Figure 7: L'itération 4

Et on choisit 2,Coût de min{1,2,3,4},Ajoute de {1,7,3,4,2}

Iteration 5:

Les sommet 1,2,3,4 et 7 relient le sommet 5,6 et avec le poids 2,2,4,1,2

On choisit le sommet 6 avec le poids plus petit 1,Coût de min{1,2,4},Ajoute de {1,7,3,4,2,6}

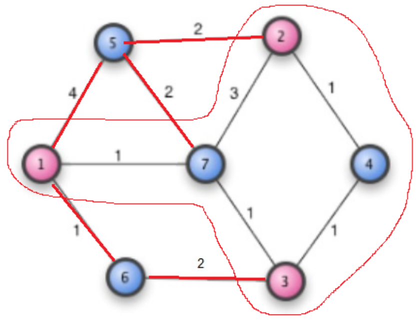


Figure 8: L'itération 5

Iteration 6:

Les sommet 1,2,3,4,6 et 7 relient le sommet 5 et avec le poids 2,2,4

On choisit le sommet 5 avec le poids plus petit 2 soit les sommets 5,7 soit 5,2

Coût de min{2,4}Ajoute de {1,7,3,4,2,6,5}

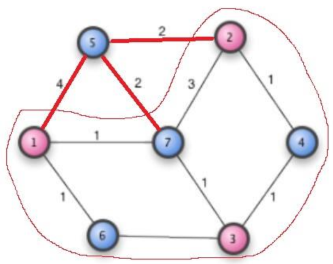


Figure 9: L'itération 6

Final:

Avec l’ajout des sommets, on peut former le graphe ci-dessous:

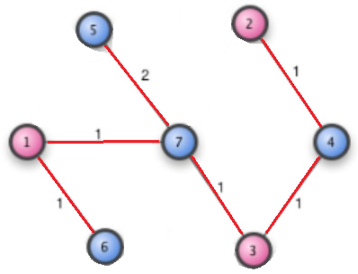


Figure 10: L'itération finale

Mais il existe un autre cas, car le sommet 5 a les deux arrêts du poids 2

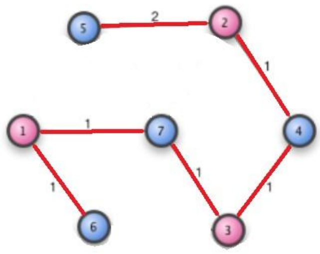


Figure 11: L'itération finale(2)

$\sum\omega=1+1+1+1+1+2=7$

Avec le poids minimal 7, on peut appliquer DFS du sommet 1 au sommet 4 si la somme du poids égale au poids minimal 7, on considere c’est une possibilité de arbre minimal et le nombre d'arrêt doit être n-1 ou n est le numéro de sommet, dans notre cas le nombre de sommet est 7 donc il faut 6 arrêts.

3.Les structures des données

On utilise les tableaux des données pour DFS et l’algo Prim, avec l’indice on peut accéder facilement dans les tableaux

```
// la stucture d'arret avec le sommet u et v , le poids w
struct arret
{
    int u,v,w;
} Arbre[Max];

int poids[Max][Max],dist[Max],visite[Max];
int sortie[Max],entree[Max];
int deja_visite[Max][Max],precedent[Max];
```

Figure 12: Les tableaux des données

4.Pseudo-code

```
Fonction prim
Entrée : SommetInitial , Status
Sortie : PoidsResultat
Début
   SommetVisité[] ← null
   PoidsResultat ← 0
   IndexIO ← 0

   Pour (i allant de 1 à nb_sommet) faire
      Si (ArretVisité[SommetInitial][i] = Status) alors
         coutAjout[i] ← poids[SommetInitial][i]
      Sinon
         coutAjout[i] ← ∞
      Fin si

      SommetPrecedent[i] ← SommetInitial
   Fin pour

   SommetVisité[SommetInitial] ← 1

   Pour (i allant de 1 à nb_sommet-1) faire
      ArretPlusPetite ← ∞

      Pour (j allant de 1 à nb_sommet) faire
         Si ( (coutAjout[j] < ArretPlusPetite) ∧ (SommetVisité[j] = 0) ) alors
            ArretPlusPetite ← coutAjout[j]
            SommetInitial = j
         Fin si
      Fin pour

   Si (ArretPlusPetite = ∞) alors
      renvoyer -1
   Fin si

   SommetVisité[SommetInitial] ← 1
   PoidsResultat ← PoidsResultat + coutAjout[SommetInitial]
   sortie[IndexIO] ← SommetPrecedent[SommetInitial]
   entrée[IndexIO++] ← SommetInitial

   Pour (k allant de 1 à nb_sommet) faire
      Si ( (poids[SommetInitial][k] < coutAjout[k]) ∧ (SommetVisité[k] = 0) ∧
(ArretVisité[SommetInitial][k] = Status)) alors
         coutAjout[k] ← poids[SommetInitial][k]
         SommetPrecedent[k] = SommetInitial
      Fin si
   Fin pour
```

```
    renvoyer PoidsResultat
Fin

Fonction dfs
Entrée : ArretChoisi , nb_arret_choisi, poids
Sortie : null

Début
    Si (poids > PoidsMinimum || nb_arret_choisi >= nb_sommet) alors
        renvoyer
    Fin si

    Si (ArretChoisi = nb_arret) alors
        Si ((nb_arret_choisi = nb_sommet-1) ∧ (PoidsResultat = poids_mini)) alors
            Si (prim(1,1) = poids_mini) alors
                print(nb_sommet-1)
            Fin si
        Fin si
        renvoyer
    Fin si

    arret cpt ← Arbre[ArretChoisi];

    deja_visite[cpt.u][cpt.v] ← deja_visite[cpt.v][cpt.u] ← 1
    dfs(ArretChoisi+1,nb_arret_choisi+1,poids+cpt.w)

    deja_visite[cpt.u][cpt.v] ← deja_visite[cpt.v][cpt.u] ← 0
    dfs(ArretChoisi+1,nb_arret_choisi,poids)
Fin

Fonction main
Entrée : null
Sortie : 0

Début
    print("Taper le nombre du sommet et d'arret")
    Boucle (scan nb_sommet, nb_arret) faire
        init()
        print("Taper les points u et v,puis le poids de cet arret")
        Pour (i allant de 0 à nb_arret) faire
            scan sommetU,sommetV,poidsW_entre_UV
            Si (poidsW_entre_UV < poids[u][v]) alors
                poids[u][v] ← poids[v][u] ← w
            Fin si

            Arbre[i].u ← u
            Arbre[i].v ← v
```

```
    Arbre[i].w ← w
  Fin Pour

  temp1 ← clock()
  poids_mini ← prim(1,0)
  temp2 ← clock()
  print("Le poids minimal(somme de w) est %d\n",poids_mini)
  print("Le temps de Prim est %f ms\n",(float)(temp2-temp1)*1000/CLOCKS_PER_SEC)

  Si (poids_mini = -1) alors
    print("Il n'y a pas d'arbre minimal, retaper\n")
  Si non
    temp3 ← clock()
    dfs(0,0,0)
    temp4 ← clock()
    print("Le temps de DFS est %f
ms\n",(float)(temp2-temp1)*1000/CLOCKS_PER_SEC)
  Fin si
 Fin Boucle
 renvoyer 0
Fin
```

5.Analyse de la complexité

On souhaite de réaliser l'algorithme par la list d'adjacence

D'après les 2-dimension de tableau, on considère comme un matrice qui comporte n*n éléments, donc la complexité est $O(n^2)$

DFS a la complexité de O(N+M) (N est le nombre du sommet ,M est le nombre d'arrêts).

6.Description de fonction

On s'intéresse à l'algorithme de Prim.

D'abord on initialise le tableau de balise ou marque pour éviter d'être la boucle. Le sommet 1 est considéré comme le sommet initial et on retire tous les poids des arrêts associés et on remarque que le sommet initial est visité d'éviter d'être la boucle.

Alors on parcourt tous les sommets pour trouver l'arbre couvrant minimum, dans la boucle de parcours, on cherche les sommets qui correspond au plus petit arrêt et remarque que le sommet est visite.Dans la dernière boucle, on retire les poids d'arrêt qui correspond à ce sommet.

7. Résultats obtenus

```
kzel@ubuntu:~/Desktop$ ./a.out
Taper le nombre du sommet et d'arret
7 10
Taper les sommets u et v,puis le poids de cet arret
1 5 4
5 2 2
2 4 1
4 3 1
3 6 2
6 1 1
1 7 1
5 7 2
7 2 3
7 3 1
Le poids minimal(somme de w) est 7
Le temps de Prim est 0.014000 ms
```

Figure 13: saisir des données

Apres saisir le nombre de sommets et arêtes, puis on saisit les deux sommets et le poids d'arrêt entre les 2 sommets.
Soit avec le poids 2 entre sommet 3 et 6, d'après l'analyse manuellement, on a 2 possibilités.
On a bien les résultats qu'on veut

```
7 3 1
Le poids minimal(somme de w) est 7
Le temps de Prim est 0.015000 ms

Le 1e arbre minimum:
Le sommet 1 -> 6 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 1 -> 7 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 3 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1, Le sommet 4 -> 2 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1, Le sommet 4 -> 2 avec poids 1, Le sommet 2 -> 5 avec poids 2,

Le 2e arbre minimum:
Le sommet 1 -> 6 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 1 -> 7 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 3 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1, Le sommet 4 -> 2 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1, Le sommet 4 -> 2 avec poids 1, Le sommet 7 -> 5 avec poids 2,
Le temps de DFS est 0.015000 ms
```

Figure 14: Le résultat obtenu

Mais le temps d'exécuter Prim et DFS est le même.
Si on essaie le poids 1

```
Le 1e arbre minimum:
Le sommet 1 -> 6 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 6 -> 3 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 6 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 6 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1, Le sommet 4 -> 2 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 6 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1, Le sommet 4 -> 2 avec poids 1, Le sommet 1 -> 7 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 6 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1, Le sommet 4 -> 2 avec poids 1, Le sommet 1 -> 7 avec poids 1, Le sommet 2 -> 5 avec poids 2,

Le 2e arbre minimum:
Le sommet 1 -> 6 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 6 -> 3 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 6 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 6 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1, Le sommet 4 -> 2 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 6 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1, Le sommet 4 -> 2 avec poids 1, Le sommet 3 -> 7 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 6 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1, Le sommet 4 -> 2 avec poids 1, Le sommet 3 -> 7 avec poids 1, Le sommet 2 -> 5 avec poids 2,

Le 3e arbre minimum:
Le sommet 1 -> 7 avec poids 1,
Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 3 avec poids 1,
Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1,
Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1, Le sommet 4 -> 2 avec poids 1,
Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1, Le sommet 4 -> 2 avec poids 1, Le sommet 3 -> 6 avec poids 1,
Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1, Le sommet 4 -> 2 avec poids 1, Le sommet 3 -> 6 avec poids 1, Le sommet 2 -> 5 avec poids 2,

Le 4e arbre minimum:
Le sommet 1 -> 6 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 1 -> 7 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 3 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1, Le sommet 4 -> 2 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1, Le sommet 4 -> 2 avec poids 1, Le sommet 2 -> 5 avec poids 2,

Le 5e arbre minimum:
Le sommet 1 -> 6 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 6 -> 3 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 6 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 6 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1, Le sommet 4 -> 2 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 6 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1, Le sommet 4 -> 2 avec poids 1, Le sommet 1 -> 7 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 6 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1, Le sommet 4 -> 2 avec poids 1, Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 5 avec poids 2,
```

```
Le 6e arbre minimum:
Le sommet 1 -> 6 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 6 -> 3 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 6 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 6 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1, Le sommet 4 -> 2 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 6 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1, Le sommet 4 -> 2 avec poids 1, Le sommet 3 -> 7 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 6 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1, Le sommet 4 -> 2 avec poids 1, Le sommet 3 -> 7 avec poids 1, Le sommet 7 -> 5 avec poids 2,

Le 7e arbre minimum:
Le sommet 1 -> 7 avec poids 1,
Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 3 avec poids 1,
Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1,
Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1, Le sommet 4 -> 2 avec poids 1,
Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1, Le sommet 4 -> 2 avec poids 1, Le sommet 3 -> 6 avec poids 1,
Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1, Le sommet 4 -> 2 avec poids 1, Le sommet 3 -> 6 avec poids 1, Le sommet 7 -> 5 avec poids 2,

Le 8e arbre minimum:
Le sommet 1 -> 6 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 1 -> 7 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 3 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1, Le sommet 4 -> 2 avec poids 1,
Le sommet 1 -> 6 avec poids 1, Le sommet 1 -> 7 avec poids 1, Le sommet 7 -> 3 avec poids 1, Le sommet 3 -> 4 avec poids 1, Le sommet 4 -> 2 avec poids 1, Le sommet 7 -> 5 avec poids 2,
Le temps de DFS est 0.017000 ms
```

Figure 15: Le résultat obtenu(2)

On a 8 résultats différentes avec le poids minimal 7

8.Les difficultés rencontrées

A cause du covid-19, la communication est difficile pendant le confinement. L'Intégration de 2 algorithmes est un peu compliqué

9.Bibliographie

[Algorithme de Prim — Wikipédia (wikipedia.org)]
[Algorithme de parcours en profondeur — Wikipédia (wikipedia.org)]
[Determination of Minimal Spanning Tree YouTube]
[python - All minimum spanning trees implementation - Stack Overflow]