

# **Devoir 3**

BLAIS REGOUT, Lucien - 18073291

SAVOIE, Olivier - 18114274

IFT436 - Algorithmes et structures de données

Faculté des Sciences  
Université de Sherbrooke

Présenté à Pr BLONDIN, Michael

Jeudi, 10 octobre 2019

## Question 1

**a)**

$V$  = Les bassins du manège aquatique (sommets).

$E$  = Les corridors entre les bassins, potentiellement des glissades (arêtes).

$\mathcal{G}$  = Manège aquatique est dirigé.

**b)**

$\mathcal{G}$  ne peut pas être un cycle dû aux contraintes stipulées. Cette contrainte, ci-dessous, en témoigne.

| Il n'y a *pas* de corridor passant d'un bassin vers lui-même.

Il est impossible qu'un cycle soit établi avec un seul bassin considérant qu'il y a aucun corridor qui part d'un dit bassin en arrivant dans ce même bassin.

Finalement, cette seconde contrainte, implique que, dû à l'inclinaison, l'un ne peut pas revenir au bassin précédent. Cela implique qu'il y a aucun cycle possible et que de ce fait, le manège aquatique  $\mathcal{G}$  est acyclique et dirigé.

| Puisque les bassins sont situés de plus en plus bas, le long d'une colline, si on peut atteindre un bassin  $j$  à partir d'un bassin  $i$ , alors on ne peut pas atteindre le bassin  $i$  à partir du bassin  $j$ .

**c)**

Si le graphe  $\mathcal{G}$  est considéré non dirigé, il y a plusieurs arrangements d'arêtes qui rendrait  $\mathcal{G}$  cyclique et pour lesquels  $|E| > |V| - 1$ . Ainsi, dans la majorité des cas,  $\mathcal{G}$  n'est pas un arbre, car il ne respecte pas les propriétés essentielles pour en être un. Toutefois, il y a toujours la possibilité qu'aucun cycle, simple ou non, ne s'y retrouve et que le nombre d'arête soit équivalent au nombre de sommet  $-1$ . De sorte que les propriétés suivantes soit respectées.

—  $\mathcal{G}$  est *connexe* et  $|E| = |V| - 1$

**d)**

$$\min = n - 1, \quad \forall n \geq 1$$

Dans le cas minimal, les arêtes placées représenteront un graphe  $\mathcal{G}$  qui sera forcément un arbre, respectant les contraintes, ainsi que l'explication élaborée à la lettre précédente, **c**).

$$\max = \sum_{i=1}^n n - i = \sum_{i=1}^n n - \sum_{i=1}^n i = n^2 - \frac{n}{2}(n+1) = \frac{n^2 - n}{2}$$

Advenant que chacun des bassins présents se voient attribués un corridor vers chacun des bassins soudjacents, l'équation maximal reflètera la composition du chemin le plus long.

**e)**

Le **bassin de départ** est le seul bassin qui ne vas jamais se retrouver à droite dans une paire de sommet, donc qui ne sera jamais  $j$  dans un couple  $(i, j)$ .

Le **bassin d'arrivée** : est le seul bassin qui ne vas jamais se retrouver à gauche dans une paire de sommet, donc qui ne sera jamais  $i$  dans un couple  $(i, j)$ .

f)

---

**Algorithme 1** : Calcul de la séquence au temps maximal passé dans une attraction  $\mathcal{G}$ 

---

**Entrées** : Tableau associatif  $S : [n] \times [n] \rightarrow \mathbb{N} > 0$ .

**Résultat** : Le temps maximal.

```
1  $MaxSommet \leftarrow [n]$ 
2  $SommeTemps \leftarrow 0$ 
3  $Enfant \leftarrow 0$ 
4 parcours ( $x$ ) :
5   si  $x$  non marquée alors
6     pour  $v \in S[x]$  faire
7       parcours ( $v$ )
8        $SommeTemps \leftarrow MaxSommet[Enfant] + c[x, v]$ 
9       si  $SommeTemps > MaxSommet[x]$  alors
10         $MaxSommet[x] \leftarrow SommeTemps$ 
11     marquer  $x$ 
12    $Enfant \leftarrow x - 1$ 
13 parcours ( $u$ )
14 retourner  $MaxSommet[0]$ 
```

---

Dans cet algorithme, un sommet est marqué uniquement lorsque tous ses voisins ont été marqués.

Le fait d'utiliser un tableau dans lequel se trouve le temps maximal pour chaque sommet, il n'est plus nécessaire de faire tous les chemins possibles. On a juste besoin d'arrêter, dès qu'on voit un sommet marqué et prendre la valeur qui lui est attribuée.

Cet algorithme se rappelle récursivement sur chacun des enfants adjacents, jusqu'à ce qu'on rencontre un sommet qui a déjà été marqué.

Ainsi, l'on peut en déduire que son temps d'exécution sera, dans le pire cas,  $\mathbf{O}(E+V)$  ce qui respecte la contrainte comme quoi l'algorithme doit être de  $\mathbf{O}(n + m)$ .

g)

---

**Algorithme 2 :** Calcul de la séquence au temps maximal passé dans une attraction  $\mathcal{G}$ , donné

5 remontées arbitrairement choisies.

---

**Entrées :** Liste d'adjacence, tableau associatif  $c : [n] \times [n] \rightarrow \mathbb{N} > 0$  et tableau associatif

$r : [n] \rightarrow \mathbb{N} > 0$

**Résultat :** Le temps maximal.

1  $MaxSommet \leftarrow [n]$

2  $SommeTemps \leftarrow 0$

3  $Enfant \leftarrow 0$

4 **parcours** ( $x$ ) :

5     **si**  $x$  non marquée **alors**

6         **pour**  $v \in S[x]$  **faire**

7             **parcours** ( $v$ )

8              $SommeTemps \leftarrow MaxSommet[Enfant] + c[x, v]$

9              $MaxSommet[x - 1] \leftarrow MaxSommet[x - 1] + SommeTemps$

10           $MaxSommet[x - 1] \leftarrow MaxSommet[x - 1] + 5 \cdot r[x]$

11           $Enfant \leftarrow x - 1$

12 **parcours** ( $u$ )

13 **retourner**  $MaxSommet[0]$

---

Ici, il fera exactement comme l'algorithme précédent, à l'exception que lorsqu'il se rendra au plus long parcours précédent le bassin final, et se 5 fois. La 5e fois, le sera possiblement différent mais restera cependant couverts par l'algorithme.

## Question 2

a)

---

**Algorithme 3 :** Tri d'une séquence et calcul des valeurs modales

---

**Entrées :** Une séquence  $S$  non vide d'éléments comparables.

**Résultat :** Les valeurs modales  $m$  trouvées dans la séquence  $S$

---

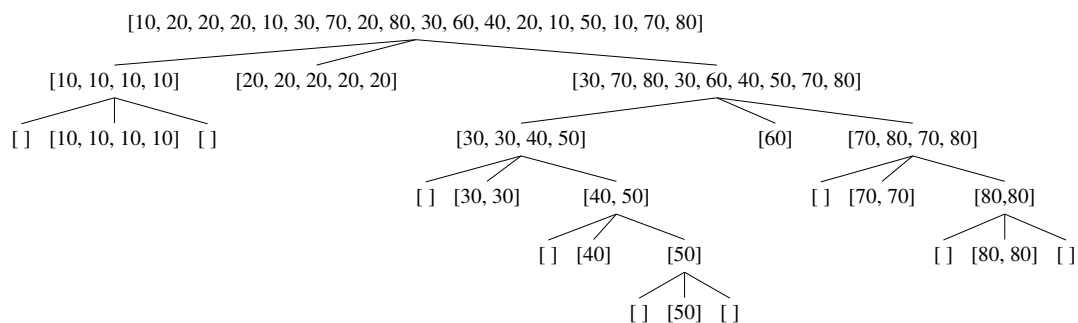
```

1   $mode \leftarrow []$ 
2   $modeQty \leftarrow 0$ 
3   $\text{trouverMode}(x)$  :
4      si  $|x| = 0$  alors
5          retourner  $mode$ 
6      sinon
7           $pivot \leftarrow \text{mediane}(x)$ 
8           $gauche \leftarrow [x \in S : x < pivot]$ 
9           $centre \leftarrow [x \in S : x = pivot]$ 
10          $droite \leftarrow [x \in S : x > pivot]$ 
11         si  $|centre| = modeQty$  alors
12             ajouter  $pivot$  à  $mode$ 
13         si  $|centre| > modeQty$  alors
14              $modeQty \leftarrow |centre|$ 
15              $mode \leftarrow [pivot]$ 
16         sinon
17              $\text{trouverMode}(gauche)$ 
18              $\text{trouverMode}(droite)$ 
19 retourner  $\text{trouverMode}(S)$ 

```

---

b)



### Question 3

a)

Donnez un algorithme qui détermine si un graphe non dirigé est un arbre.

---

**Algorithme 4 :** Détermination de la propriété d'arbre d'un graphe.

---

**Entrées :** Un graphe  $\mathcal{G}$  tel que  $\mathcal{G} = (V, E)$  est non dirigé et un sommet  $u \in V$

**Résultat :** Si le graphe est un arbre, ou non.

```
1  $Visites \leftarrow 0$ 
2 visiter( $x$ ) :
3   si  $x$  est marqué alors
4     retourner  $Non\_Arbre$ 
5   marquer  $x$ 
6    $Visites \leftarrow Visites + 1$ 
7   pour tous voisins de  $x \in V : x \rightarrow y$  faire
8     visiter( $y$ )
9 visiter( $u$ )
10 si  $Visites \neq |V|$  alors
11   retourner  $Non\_Arbre$ 
12 sinon
13   retourner  $Est\_Arbre$ 
```

---

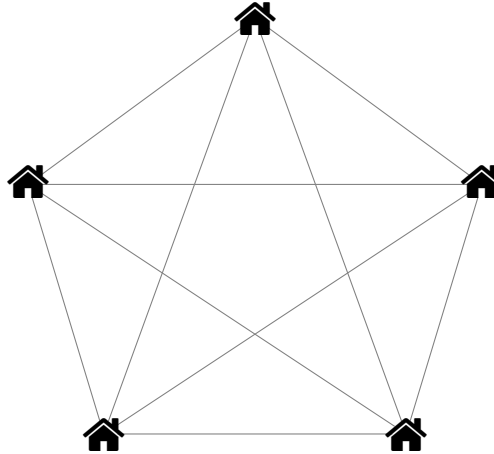
Cet algorithme parcourt le graphe au complet. Si un sommet est visité plus d'une fois, c'est cyclique. Cela est uniquement véridique lorsqu'on empêche de visiter le noeud précédents en visitant tous les voisins immédiats.

Deplus, tous les sommets peuvent être visiter moins de deux fois et tout de même il pourrait y manquer certains sommets, d'ou la vérification si le nombre de visites est bien égale a la somme des sommets dans le graphes !

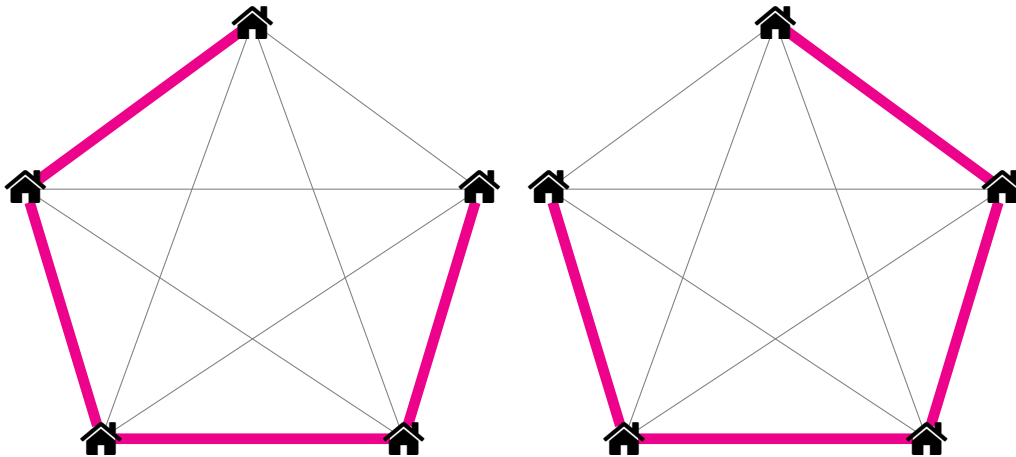
b)

Démontrez que le graphe complet  $\mathcal{G}_n$  possède au moins  $2^n$  arbres couvrants,  $\forall n \in \mathbb{N} > 3$ .

L'énoncé stipule que pour  $n = 4$  où  $n$  est le nombre de gîtes, il y a  $2^4 = 16$  réseaux possibles. Si nous ajoutons une autre gîte, nous obtenons ce dernier, à 5 gîtes.



Afin de prouver qu'il y a au minimum  $2^n$  chemins possibles, utilisons un sous-graphes  $\mathcal{H}$  de  $\mathcal{G}$ . En prenant le sous-graphe  $\mathcal{H}$  possédant 4 sommets (gîtes), il est possible de savoir qu'il y a  $2^n$  chemins possibles. De ce fait, on peut voir, que ce nombre est doublé lorsque le graphe est complet (donc avec 5 sommets). En effet, pour chacune des chemins en  $n = 4$ , il y a 2 chemins possibles lorsque  $n = 5$ .

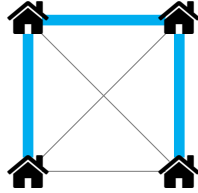


Donc pour  $n = 5$ , le nombre de chemin minimum possible égale  $2^4 \cdot 2^1 = 2^5$  chemins possible. Donc il est possible de dire que pour un graphe complet, il existe au minimum  $2^n$  arbre couvrant  $n \in \mathbb{N} \geq 4$ .



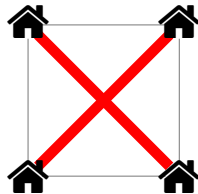
c)

Donnez un exemple de points où un arbre couvrant minimal entre ces points est plus long qu'un arbre couvrant minimal avec un nouveau point ajouté (une halte).



$$3 * 1 = 3u$$

Dans cette image, on retrouve un arbre couvrant minimal qui couvre 4 sommets d'un graph  $\mathcal{G}$  à 4 sommets. Assumant qu'il y a 1 unité ( $u$ ) de distance entre les coins, du carré formé, adjacents, le graph aurait une longueur suivant l'équation ci-dessus.



$$4\left(\frac{1}{2}\right)(1^2 + 1^2) = 2^{\frac{1}{2}} = 2.8284 \dots u$$

Maintenant, ces gites sont équidistant de cette halte, sont à la même position que le graphe précédent et la distance totale de chacun des gites à la halte va comme suit.