Devoir 3

BLAIS REGOUT, Lucien - 18073291
SAVOIE, Olivier - 18114274

IFT436 - Algoritmes 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 représenterons un graphe $\mathcal{G}$ qui sera forcément un arbre, respectant les contraintes, ainsi que l'explication élaborée à la lettre précédente, **c)**.

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

Advennant 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 : Manège $\mathcal{G}$ tel que $\mathcal{G}= (V,E)$ et sommet initial u tel que u appartien à V.

Résultat : Une séquence S de bassins de temps maximal dans le manège $\mathcal{G}$.

```
S ← []
Ttot ← 0
Parent ← u
parcours (x) :
    SPrime ← S
    Temps ← Ttot
    si x non marquée alors
        Temps ← Temps + c[Parent, x]
        marquer x
        ajouter x à SPrime
        si Temps > Ttot alors
            Ttot ← Temps
            S ← SPrime
        pour u ∈V : x → y faire
            Parent ← x
            parcours (y)
parcours (u)
retourner S
```

Grâce aux marqueurs, l'algorithmes ne visite qu les éléments qui ne furent pas précédement visités. De plus, ce dernier se rappelle récursivement sur chacun des enfants soudjacents. Ainsi, l'on peut en déduire que son temps d'exécution sera, dans le pire cas, $\mathcal{O}(|E| + |V|)$ ce qui respecte la contrainte de $\mathcal{O}(n + m)$.

g)

Algorithme 2 : Calcul de la séquence au temps maximal passé dans une attraction $\mathcal{G}$, donné 5 remontées

Entrées : Manège $\mathcal{G}$ tel que $\mathcal{G}= (V,E)$, que le sommet initial u tel que $u \in V$, que $v \in V$, ainsi que v est le bassin finale.

Résultat : Une séquence S de bassins, de temps maximal dans le manège $\mathcal{G}$, considérant que l'utilisateur peut remonter au bassin initial un maximum de 5 fois.

$S \leftarrow [\,]$
$Ttot \leftarrow 0$
$Parent \leftarrow u$
$Ascension \leftarrow 5$
`parcours` *(x)* **:**
    $SPrime \leftarrow S$
    $Temps \leftarrow Ttot$
    si x *non marquée* **alors**
        $Temps \leftarrow Temps + c[Parent, x]$
        marquer x
        ajouter x à $SPrime$
        si $Temps > Ttot$ **alors**
            $Ttot \leftarrow Temps$
            $S \leftarrow SPrime$
        si *prochain voisin de* $x = v$ **alors**
            $Ascension \leftarrow Acension - 1$
            retirer les marqueurs
            $Parent \leftarrow S[0]$
            `parcours`($Parent$)
        sinon
            pour $u \in V : x \rightarrow y$ **faire**
                $Parent \leftarrow x$
                `parcours`(y)
`parcours`(u)
retourner S

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 couvers par l'algorithme.

Question 2

a)

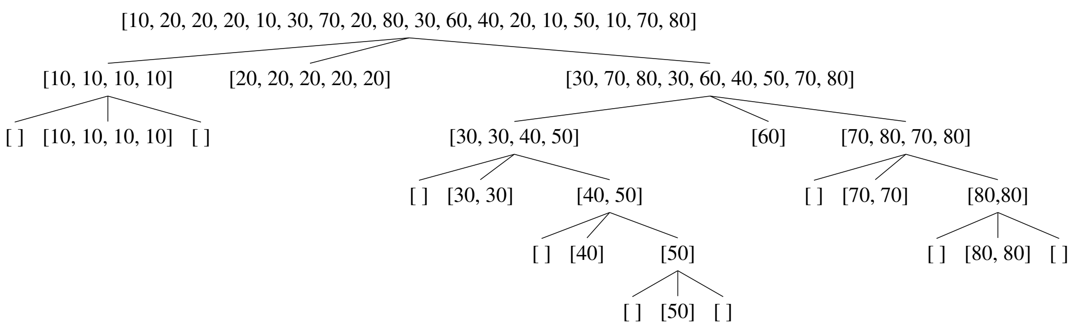
Algorithme 3 : Tri d'une séquence et calcul des valeurs modales

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

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

$mode \leftarrow [\,]$
$modeQty \leftarrow 0$
`trouverMode` (x) **:**
  si $|x| = 0$ **alors**
    retourner $mode$
  sinon
    $pivot \leftarrow$ `mediane` (x)
    $gauche \leftarrow [x \in S : x < pivot]$
    $centre \leftarrow [x \in S : x = pivot]$
    $droite \leftarrow [x \in S : x > pivot]$
    si $|centre| = modeQty$ **alors**
      ajouter $pivot$ à $mode$
    si $|centre| > modeQty$ **alors**
      $modeQty \leftarrow |centre|$
      $mode \leftarrow [pivot]$
    sinon
      `trouverMode` $(gauche)$
      `trouverMode` $(droite)$
retourner `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 graph $\mathcal{G}$ tel que $\mathcal{G}= (V,E)$ est non dirié et un sommet $u \in V$

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

$Visites \leftarrow 0$
`visiter`(x) **:**
    si x *est marqué* **alors**
        retourner Non_Arbre
    marquer x
    $Visites \leftarrow Visites + 1$
    pour tous *voisins de* $x \in V : x \rightarrow y$ **faire**
        `visiter`(y)
`visiter`(u)
si $Visites \neq ||$ **alors**
    retourner Non_Arbre
sinon
    retourner Est_Arbre

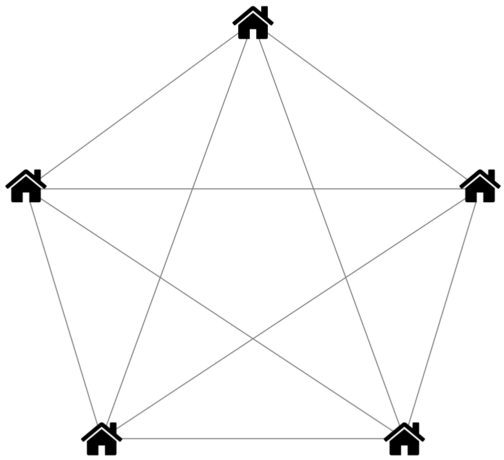
Cet algorithme parcours le graph au complet. Si un sommet est visité plus d'une fois, c'est cyclique. Cela est uniquement véridicte lorsqu'on empêche de visiter le noeud précédents en visitant tous les voisins immédiats.

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 !

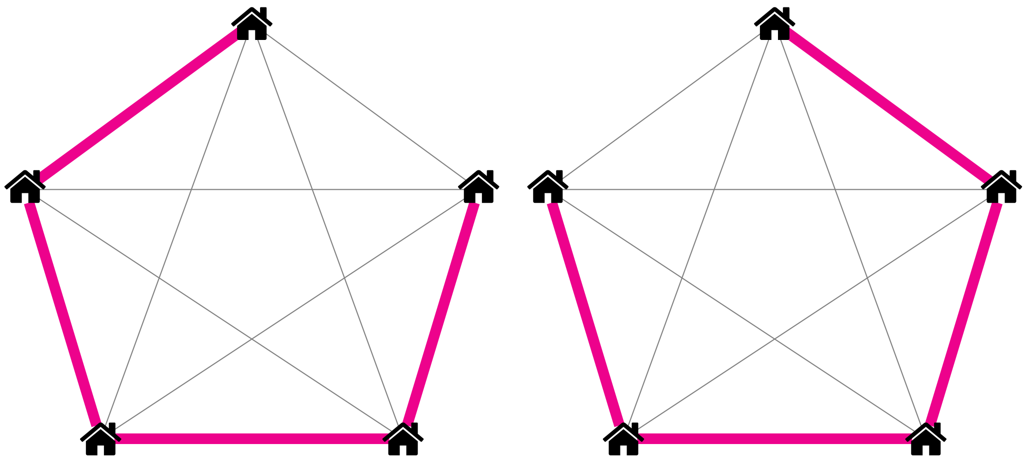
b)

> **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 gites, il y a $2^4 = 16$ réseaux possibles. Si nous ajoutons une autre gites, nous obtenons ce dernier, à 5 gites.



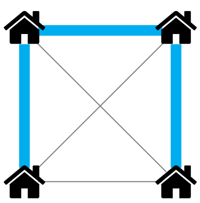
Afin de prouver qu'il y a au minimum 2^n chemins possibles, utilisons un sous-graphes $\mathcal{H}$ de $\mathcal{G}$. En prenant le sous-graphe $\mathcal{H}$ possedant 4 sommets (gites), il est possible de savoir qu'il y a 2^n chemins possibles. De ce fait, on peut voir, que ce nombre est doublé lorsque 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 $2^4 * 2^1 = 2^5$ 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)

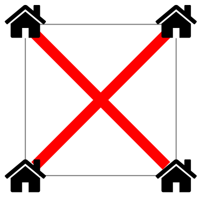
> **Donnez un exemple de points où un arbre couvrant minimal entre ces points est plus long qu'un arbre couvrant minimal avec un nouveau point ajouté (une halte).**



$$3 * 1 \ = \ 3u$$

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, adjacants, le graph aurait une longueur suivant l'équation ci-dessus.



$$4(\tfrac{1}{2})(1^2 + 1^2) \ =$$
$$2^{\frac{1}{2}} \ = \ 2.8284\ldots u$$

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.