Devoir 3

BLAIS REGOUT, Lucien
SAVOIE, Olivier

IFT436 - Algoritmes et structures de données

Faculté des Sciences
Université de Sherbrooke

Présenté à Pr BLONDIN, Michael

Jeudi, 10 octobre 2019

Première Question

a)

V = Les bassins du manège aquatique (sommets). ***E*** = Les corridors entre les bassins, potentiellement des glissades(arêtes). ***G*** = Manège aquatique, est dirigé, du à l'élevation. Élévation dénoté de l'énoncé suivant :

> 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.

b)

G ne peut pas être un cycle dû contraintes fournis ci-dessous. Grâce à cette première contraites plus bas, 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, la 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 ***G*** est acyclique et dirigé.

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

> 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 ***G*** est considéré non dirigé, il y a plusieurs arrangements d'arêtes qui rendrait ***G*** cyclique. Toutefois, il y a toujours la possibilité qu'aucun cycle, simple ou non, ne soit présent et que le nombre d'arête soit équivalent au nombre de sommet -1. De sorte que les propriétés suivantes soit respectées.

- ***G*** est *connexe*
- ***G*** est *acyclique*
- $|\boldsymbol{E}| = |\boldsymbol{V}|$ **YALL C'EST PAS BON!!!!!**

d)

$$\forall n \geq 1$$

$$min = n - 1$$

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

e)

Bassin départ : De toutes les paires de sommets composants les arêtes, le sommet qui se retrouvera uniquement en index 0 de la paire, à la gauche, qui est donc toujours le sommet de départ, sera donc considéré comme sommet de départ.

Bassin d'arrivée : Suivre le même résonnement, mais pour le sommet qui se retrouve toujours en index 1 de la paire et qui est donc toujours le bassin d'arrivée dans toutes les paires, sera donc considéré comme sommet d'arrivée.

f)

Algorithme 1 : Calcul de la séquence qui possède un temps maximal passé dans une attraction ***G***

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

Résultat : Une séquence S de bassins de temps maximal dans le manège ***G***.

```
1  S ← []
2  Ttot ← 0
3  Parent ← u
4  parcours (x) :
5      SPrime ← S
6      Temps ← Ttot
7      si x non marquée alors
8          Temps ← Temps + c[Parent, x]
9          marquer x
10         ajouter x à SPrime
11         si Temps > Ttot alors
12             Ttot ← Temps
13             S ← SPrime
14         pour u ∈V : x → y faire
15             Parent ← x
16             parcours(y)
17 parcours(u)
18 retourner S
```

g)

Algorithme 2 : Calcul de la séquence qui possède un temps maximal passé dans une attraction ***G***, donné 5 remontées

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

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

```
1  S ← [ ]
2  Ttot ← 0
3  Parent ← u
4  Ascension ← 5
5  parcours (x) :
6      SPrime ← S
7      Temps ← Ttot
8      si x non marquée alors
9          Temps ← Temps + c[Parent, x]
10         marquer x
11         ajouter x à SPrime
12         si Temps > Ttot alors
13             Ttot ← Temps
14             S ← SPrime
15         si prochain voisin de x = v alors
16             Ascension ← Acension − 1
17             retirer les marqueurs
18             Parent ← S[0]
19             parcours(Parent)
20         sinon
21             pour u ∈V : x → y faire
22                 Parent ← x
23                 parcours(y)
24 parcours(u)
25 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.

Seconde Question

a)

To be completed

b)