

Devoir 3

BLAIS REGOUT, Lucien SAVOIE, Olivier

IFT436 - Algorithmes 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'élévation. 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 contraintes 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*

— $|E| = |V|$ **YALL C'EST PAS BON!!!!**

d)

$$\begin{aligned}\forall n &\geq 1 \\ \min &= n - 1 \\ \max &= \sum_{i=1}^n n - i = \sum_{i=1}^n n - \sum_{i=1}^n i = n^2 - \frac{n}{2}(n + 1) = \frac{n^2 - n}{2}\end{aligned}$$

e)

Bassin départ : De toutes les paires de sommets composant 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 raisonnement, 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 $G = (V, E)$ et sommet initial *u* tel que *u* appartien à *V*.

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

```
1  $S \leftarrow []$ 
2  $T_{tot} \leftarrow 0$ 
3  $Parent \leftarrow u$ 
4 parcours (x) :
5    $SPrime \leftarrow S$ 
6    $Temps \leftarrow T_{tot}$ 
7   si x non marquée alors
8      $Temps \leftarrow Temps + c[Parent, x]$ 
9     marquer x
10    ajouter x à SPrime
11    si  $Temps > T_{tot}$  alors
12       $T_{tot} \leftarrow Temps$ 
13       $S \leftarrow SPrime$ 
14    pour  $u \in V : x \rightarrow y$  faire
15       $Parent \leftarrow x$ 
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 $G = (V, E)$, 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 G , considérant que l'utilisateur peut remonter au bassin initial un maximum de 5 fois.

```

1  $S \leftarrow []$ 
2  $T_{tot} \leftarrow 0$ 
3  $Parent \leftarrow u$ 
4  $Ascension \leftarrow 5$ 

5 parcours ( $x$ ) :
6    $S_{Prime} \leftarrow S$ 
7    $Temps \leftarrow T_{tot}$ 
8   si  $x$  non marquée alors
9      $Temps \leftarrow Temps + c[Parent, x]$ 
10    marquer  $x$ 
11    ajouter  $x$  à  $S_{Prime}$ 
12    si  $Temps > T_{tot}$  alors
13       $T_{tot} \leftarrow Temps$ 
14       $S \leftarrow S_{Prime}$ 
15    si prochain voisin de  $x = v$  alors
16       $Ascension \leftarrow Ascension - 1$ 
17      retirer les marqueurs
18       $Parent \leftarrow S[0]$ 
19      parcours ( $Parent$ )
20    sinon
21      pour  $u \in V : x \rightarrow y$  faire
22         $Parent \leftarrow x$ 
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 couverts par l'algorithme.

Seconde Question

a)

Im here..