Devoir maison (Partie 1)
Calculabilité et Complexité
3 Novembre 2019

ANEAS Alan et CATALDI Alexis
Master - Année 1
Spécialité informatique

Professeur de TD :
SENHAJI Mohammed

Professeur en amphithéâtre :
MUSCHOLL Anca

Sommaire

I.1 Définition du problème Distance commune……………………………..……………………3

I.2 Cas particulier acyclique
2.1………………………………………………………………………………………………………3
2.2………………………………………………………………………………………………………3
2.3 (Bonus)...…………………………………………………………………………...…………3

I.3 Réduction de Distance commune vers SAT
3.1.a……………………………………………………………………………………………………4
3.1.b……………………………………………………………………………………………………4
3.1.c……………………………………………………………………………………………………4
3.1.d……………………………………………………………………………………………………5
3.1.e……………………………………………………………………………………………………5
3.2.a……………………………………………………………………………………………………5
3.2.b……………………………………………………………………………………………………5
3.2.c……………………………………………………………………………………………………5
3.2.d……………………………………………………………………………………………………5
3.3 (Bonus)……………………………………...…………………………………………...5

Partie 1 : Exercice 1

1)
En entrée, il nous est donné un chemin par graphe.
Il suffit de vérifier que chaque chemin a la même taille, et que chaque chemin est valide, en vérifiant que chaque arête existe et qu'on ne passe pas deux fois par le même sommet.
Il faut aussi que le sommet de départ soit le sommet source, et le sommet d'arrivée soit le sommet destination de chaque graphe.

O(1) pour vérifier les tailles des chemins et que les sommets sources et destinations soient corrects.
O(n) pour parcourir le graphe en prenant les arcs mentionnés.
O(n*N) pour parcourir chaque graphe de cette façon.
Cela est bien polynomial.

Partie 1 : Exercice 2

1)
On fait le parcours du plus petit des graphes en partant du point s et l'on note la longueur des chemins arrivant à t en mettant à vrai une case d'un tableau dont l'indice correspond à la longueur du chemin.
Ensuite on parcourt les autres graphes en notant s'ils possèdent eux aussi des cases à vrai, à la fin de l'algorithme, si tous les tableaux ont une case avec le même indice à vrai alors cela veut dire que tous les graphes ont un chemin de même taille avec cette taille étant l'indice de la case.

Complexité :
Cela correspond à un parcours en profondeur de tous les graphes, soit la somme des n et m, avec n et m étant respectivement le nombre de sommets et d'arcs dans chaque graphe.
$O(\sum_{(1 \leq i \leq N)}(n+m)) = O(N^*(m+n)) = O(Nm+Nn)$, soit une complexité polynomiale en l'entrée.

2)
Dans le cas général, les boucles peuvent nous faire tourner à l'infini dans le premier graphe ce qui fausse notre tableau des tailles des chemins et surtout ne pas nous faire finir l'algorithme.

3) (Bonus) Cela pourrait tourner en boucle à l'infini.

Partie 1 : Exercice 3

1)

a)Il s'agit du nombre de sommet du plus petit des graphes.

b)"Point de départ 's'"

$\phi_1 = x_{i,0,k,s}$

"Point d'arrivée 't'"

$\phi_2 = x_{i,k,k,t}$

"Au moins 1 sommet par position"

$\phi_3 = \Lambda_{(0 \leq j \leq k)} V_{(u \in V(G))} x_{i,j,k,u}$

"Au plus 1 sommet par position"

$\phi_4 = \Lambda_{(0 \leq j \leq k)} \Lambda_{(u \in V(G))} \Lambda_{(v \in V(G),\, v \neq u)} \neg x_{i,j,k,u} \vee \neg x_{i,j,k,v}$

"Chaque sommet apparaît au plus une fois"

$\phi_5 = \Lambda_{(u \in V(G))} \Lambda_{(0 \leq j \leq k)} \Lambda_{(0 \leq j2 \leq k,\, j2 \neq j)} (\neg x_{i,j,k,u} \vee \neg x_{i,j2,k,u})$

"Chemin de taille k continue"

$\phi_6 = \Lambda_{(0 \leq j \leq k)} \Lambda_{(u \in V(G))} (x_{i,j,k,u} \wedge V_{(v \in N(u))} x_{i,j+1,k,v})$

"Distance commune"

$\phi = \Lambda_{(1 \leq j \leq N)} (\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge \phi_5 \wedge \phi_6)$

c)Supposons qu'il existe un chemin simple et valide de longueur k allant de 's' à 't' dans chacun des graphes.

Le chemin allant de 's' à 't',pour tous les graphes on a bien $x_{i,0,k,s}$ et $x_{i,k,k,t}$. Donc ϕ_1 et ϕ_2 sont satisfaisable pour tous les graphes.

Le chemin étant simple et de taille k, alors ϕ_3, ϕ_4 et ϕ_5 sont satisfaisable.

Le chemin étant continu, alors ϕ_6 est bien satisfaisable.

Alors par construction, si il existe un chemin simple et valide de k allant de 's' à 't', ϕ est donc satisfaisable.

d)Si ϕ_1 et ϕ_2 sont satisfaisable, alors il doit exister un chemin allant ‘s’ à ‘t’.
Si ϕ_3 et ϕ_4 sont satisfaisable, alors le chemin en question est de taille k.
Si ϕ_5 est satisfaisable, alors le chemin ne passe pas deux fois par le même sommet, il est donc simple.
Si ϕ_6 est satisfaisable, alors le chemin est continu.

Par construction, si ϕ est satisfaisable, alors il existe un chemin simple et continu de taille k allant de ‘s’ à ’t’ .

e)$\phi' = \bigvee_{(1 \leq k \leq kmax)} \bigwedge_{(1 \leq j \leq N)} (\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge \phi_5 \wedge \phi_6)$
où kmax correspond au k maximal défini à la question 3.1.a.

2)
a)On crée la formule pour k = 1, si elle est satisfaisable on s’arrête, sinon on teste pour k = 2 et ainsi de suite jusqu'à kmax.

b)On peut obtenir moins de variables sur l’ensemble de l’algorithme, car nous n’avons que pour les chemin de taille ‘k’ que nous vérifions et non par pour tout les ‘k’ inférieur à ‘kmax’.

c)On ne fait pas les formules pour tous les k, on s’arrête à la première satisfaisable, si il y en a une.

d)Il suffit de faire un Λ sur les formule pour passer de chemin de taille ‘k’ avec un ‘k’ fixé à un chemin de taille au plus ‘kmax’.

3) (Bonus)