

Mathématique pour informaticien I

Devoir 2

Par : Simon Gaudette-Dupré et Jonathan Thivierge

Remis à : Ismaïl Biskri

Devoir 2

Structure

Graphe	
Variables utilisées	
▪ listeSommets	Liste contenant les sommets
▪ matriceAdjacence	Matrice qui contient les arcs entre les sommets
▪ nombreSommets	Contient le nombre total de sommets
▪ nombreCycle	Contient le nombre de cycle présent dans le graphe

Sommet	
Variables utilisées	
▪ etiquette	Identifiant du sommet
▪ degrePositif	Contient le degré positif (arc incident sortant)
▪ degreNegatif	Contient le degré négatif (arc incident entrant)

Algorithmes

calculerDegréPositifNegatif()	
Paramètres	aucun
Valeur de retour	aucune
Pseudo code	
<pre>Pour chaque rangée r Pour chaque colonne c Si matriceAdjacence[r][c] égal 1 Incréments degré positif du sommet rangée r Incréments degré négatif du sommet colonne c Fin si Fin pour chaque colonne c Fin pour chaque rangée r</pre>	

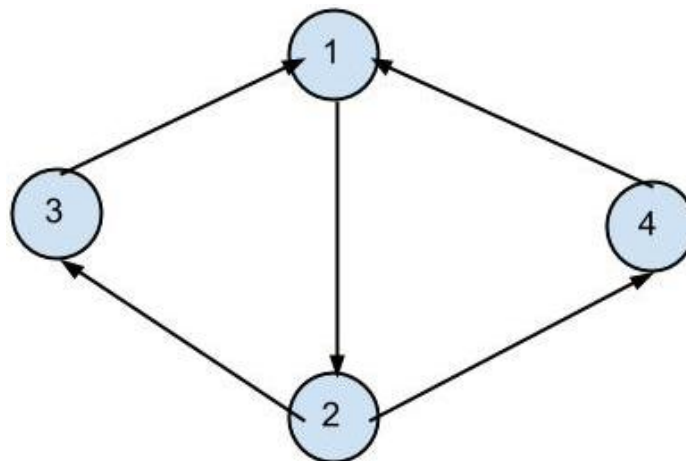
estCycleEulerien()	
Informations	Un cycle eulérien suppose que tout les sommets sont de degrés pair (graphe connexe)
Paramètres	aucun
Valeur de retour	booléen
Pseudo code	
<pre> estEulerien = vrai Pour chaque Sommet s Si (degré positif + degré négatif) % 2 != 0 estEulerien = faux Fin si Fin pour chaque sommet s retourner estEulerien </pre>	

trouverCycle()	
Informations	Fonction récursive qui parcours le graphe depuis un sommet donné et trouve tout les cycles possibles
Paramètres	sommetActuel, sommetsVisites
Valeur de retour	aucune
Pseudo code	
<pre> Si sommetsVisites contient sommetActuel Ajouter sommetActuel a sommetsVisites Afficher cycle Fin Si Sinon Ajouter sommetActuel a sommetsVisites Rangee = sommetActuel Pour chaque colonne c de matriceAdjacence Si matriceAdjacence[Rangee][c] == 1 prochainSommet = Sommet c trouverCycle(prochainSommet, sommetsVisites) Fin si Fin pour chaque colonne c Fin sinon </pre>	

validerArc()	
Paramètres	entier début, entier fin
Valeur de retour	booléen
Pseudo code	
<pre> Si (debut == fin) ou si (matrice[fin][debut] == 1) Retourner faux Sinon Retourner vrai Fin si </pre>	

validerSommetSeul()	
Paramètres	aucun
Valeur de retour	booléen
Pseudo code	
<pre> sommetSeul = vrai Pour chaque rangée r Pour chaque colonne c Si matriceAdjacence[r][c] égal 1 sommetSeul = faux Fin si Fin pour chaque colonne c Fin pour chaque rangée r Retourner sommetSeul </pre>	

Graphe considéré



Input	
Sommet départ	Sommet arrivée
1	2
2	3
3	1
2	4
4	1

Résutats

trouverCycle()
Cycles : 1 -> 2 -> 3 -> 1 1 -> 2 -> 4 -> 1 2 -> 3 -> 1 -> 2 2 -> 4 -> 1 -> 2 3 -> 1 -> 2 -> 3 3 -> 1 -> 2 -> 4 -> 1 4 -> 1 -> 2 -> 3 -> 1 4 -> 1 -> 2 -> 4 Le graphe contient 8 cycle.
estCycleEulerien()
Le graphe ne contient pas de cycle eulérien.
contientChaineEulerienne()
Le graphe contient une chaîne eulérienne.