

Réseaux informatiques

Série 6, 2019

Exercice 1 :

Complétez le code de simulateur de l'algorithme de vecteur distance qui se trouve sur moodle dans le répertoire *vecteur distance*. Seule la méthode `checkMessage()` du fichier de la classe `Pont` est à compléter.

Exercice 2 :

Faire les exercices de la page 55 du cours.

Montrez la convergence des algorithmes des pages 53 et 54 du cours pour le calcul des distances dans un graph.

Exercice 3 :

On cherche à calculer la distance entre deux nœuds x et y d'un graphe. Une idée est d'appliquer une version bidirectionnelle de l'algorithme de Dijkstra. L'algorithme de Dijkstra construit un ensemble S de nœud qui correspond aux nœuds dont la distance est déterminée (étiqueté de manière permanente)

On applique deux versions de Dijkstra, une à partir du nœud x et l'autre à partir du nœud y . La première version calcule étiquette les nœuds du graphe avec une valeur $d(j)$ qui est la distance du nœud j à x (construit l'ensemble S) et la deuxième version étiquette les nœuds du graphe avec une valeur $d'(j)$ qui est la distance du nœud j à y (construit

l'ensemble S'). On applique alternativement un pas de l'algorithme de Dijkstra pour calculer d puis d' , etc.

L'algorithme s'arrête lorsqu'un nœud k est étiqueté de manière permanente par les deux versions de l'algorithme, k appartient à S intersection S' .

Montrez que le chemin optimal entre x et y est

- Soit le chemin optimal de x à k puis de k à y .
- Soit un chemin de x à i , puis en suivant une arête (i,j) , puis de j à y avec i et j appartiennent à S et S' respectivement.

Comment on modifie l'algorithme bidirectionnel si les distances des arêtes ne sont pas symétriques ?

Dans quelles situations l'algorithme bidirectionnel est plus efficace que l'algorithme classique ?