

Algorithmes fondés sur des parcours de graphes

Exercice 1

Q 1.1

Matrice d'adjacence: n^2

Liste d'adjacence: $n \times m$

Q 1.2

Matrice d'adjacence: $O(1)$ Liste d'adjacence: $O(m)$

Matrice d'adjacence: $O(m)$

Liste d'adjacence: $O(1)$

Q 1.3

1. $O(1)$

2. $O(1)$

Exercice 2

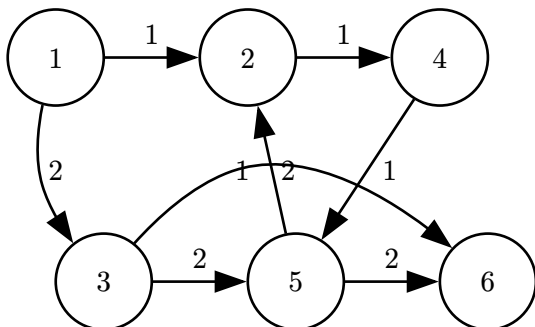
Exercice 3

Exercice 4

Exercice 5

Q 5.1

1	2	3	4	5	6
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
0	1	2	$+\infty$	$+\infty$	$+\infty$
0	1	2	2	$+\infty$	$+\infty$
0	1	2	2	4	3
0	1	2	2	3	3
0	1	2	2	3	3
0	1	2	2	3	3



Q 5.2

On rajoute des nouveaux noeuds à coté de tous les noeuds de cout 2 et on les relie au même chemin

Q 5.3

Exercice 6

Q 6.1

a	b	c	d	e	f
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
0	1	3	1	$+\infty$	$+\infty$
0	1	3	1	5	5
0	1	3	1	4	5

Q 6.2

ne change pas pour $\varepsilon \leq 1$ car on doit vérifier $d(d) + (3 - \varepsilon) \geq d(e)$

Q 6.3

Exercice 7

Exercice 8

https://en.wikipedia.org/wiki/Prim%27s_algorithm

Exercice 9

Q 9.1

Il est inchangé

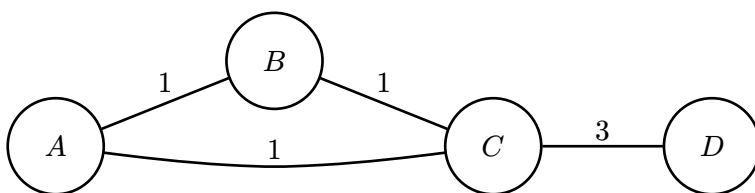
Q 9.2

Il peut changer

Exercice 10

a)

Faux



b)

Vrai, par l'absurde

c)

Vrai. Soit $e = \{u, v\}$ une arête de coût minimum. En commençant par le sommet u , l'algorithme de Prim peut sélectionner cette arête, qui est l'une des arêtes de plus petit coût du cocycle de $\{u\}$

d)

Pour la même raison \Rightarrow Vrai

e)

Faux

Exercice 11

Q 11.1

Non, on pourrait prendre l'arête F-E et avoir un coût plus petit

Q 11.2

$A \rightarrow B, B \rightarrow C, C \rightarrow D, D \rightarrow F, F \rightarrow E, E \rightarrow G$

Q 11.3

Faux

Q 11.4

Soit T un ACCM qui n'est pas un arbre couvrant de niveau de congestion minimal.

Soit x le niveau de congestion d'un arbre couvrant de niveau de congestion minimal.

Il existe dans T au moins une arête de coût strictement supérieur à x (sinon T serait un arbre couvrant de niveau de congestion minimal). Soit $\{a, b\}$ cette arête et soient C_1 et C_2 les deux arbres obtenus après suppression de $\{a, b\}$ dans T . Il existe dans le cycle de C_1 au moins une arête α de coût inférieur à x (car il existe un arbre couvrant de niveau de congestion minimal x). L'arbre formé par les arêtes de C_1 et C_2 et α est un arbre couvrant de coût strictement inférieur à T . Contradiction car T est un ACCM

Q 11.5

On peut utiliser l'algorithme de Prim, car un ACCM est de niveau de congestion minimal

Apparemment complexité $O(m \log(m))$

Q 11.6

On note g le niveau de congestion

alors $g(G) \geq g(G_i)$

Q 11.7

1.

Dépend du test de connexité:

Parcours de graphe $O(m(n + m))$

Union find $O(m \log(n))$

2.

On y sélectionne un arbre couvrant, car le niveau de congestion est nécessairement minimal

3.

Algo de la 1. + coût de trouver arbre couvrant

4.

Bon..