Graphes :

1 heure 20 minutes

Aucun document (électronique ou papier) n'est autorisé. Vous pouvez répondre en français ou en anglais à votre convenance. Les sections sont indépendantes. Toutes les réponses doivent être argumentées.

1 Arbre couvrant de poids minimum

Question 1 *Donnez la définition d'un arbre (dans le contexte des graphes).*

Un *arbre couvrant* d'un graphe G est un sous-graphe de G qui est un arbre et qui "touche" tous les sommets de G. Le poids d'un sous-graphe est la somme des poids des arêtes de ce sous-graphe.

Question 2 *Décrire (en 2-3 phrases maximum ou avec un pseudo-code) un algorithme pour calculer un arbre couvrant de poids minimum étant donné un graphe $G = (V, E)$ arête-pondérée (avec une fonction de poids $\omega : E \to \mathbb{R}^+$ sur les arêtes) en entrée. Donnez sa complexité en temps en fonction de $|E|$.*

Question 3 *Donnez un arbre couvrant de poids minimum pour le graphe pondéré décrit dans la Figure 1.*

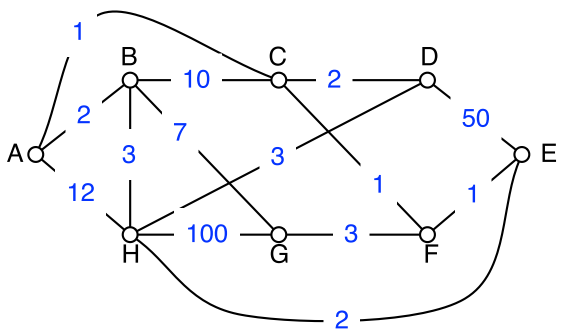


FIGURE 1 – Un graphe arête pondérée (les poids des arêtes sont en bleu).

2 Cycle Eulérien

Rappelons qu'un graphe G est *Eulerien* si il admet un tour (une séquence de sommets telle que 2 sommets consécutifs sont adjacents et dont le premier et dernier sommets sont égaux) qui passe exactement une fois par chaque arête de G.

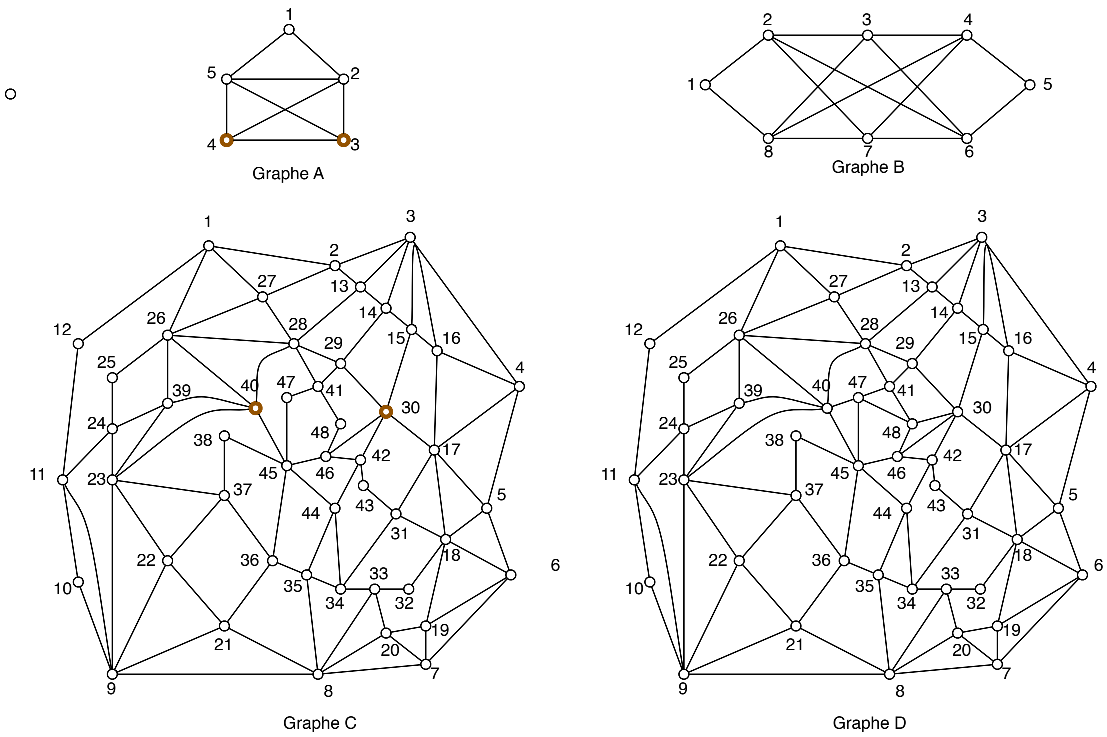


FIGURE 2 – 4 graphes A, B, C et D dont il faut déterminer s'ils sont eulériens ou non.

Question 4 *Parmi les exemples décrits dans les 4 graphes (A,B,C et D) de la Figure 2, dire ceux qui sont Euleriens*[1]*. S'ils ne le sont pas donnez un argument qui le prouve, sinon donnez un cycle Eulerien et décrire l'algorithme qui vous a permis de l'obtenir.*

3 Cycle Hamiltonien

Rappelons qu'un cycle *Hamiltonien* d'un graphe est une permutation des sommets telle que 2 sommets consécutifs sont adjacents et telle que le premier et dernier sommets sont adjacents.

Soit K_n le graphe complet à n sommets (tous les sommets sont adjacents) et $\omega : E(K_n) \to \mathbb{R}^+$ une fonction de poids sur les arêtes.

Question 5 *Décrire un algorithme (naïf) qui calcule un cycle Hamiltonien de poids minimum dans (K_n, ω). Donnez la complexité en temps de l'algorithme proposé.*

Question 6 *Que signifie que ω satisfait l'inégalité triangulaire ?*

Question 7 *Soit $c > 1$. Quelles sont les trois propriétés principales que doit satisfaire un algorithme de c-approximation pour un problème de minimisation ?*

Question 8 *Décrire une 2-approximation pour le calcul d'un cycle Hamiltonien de poids minimum dans (K_n, ω) si ω satisfait l'inégalité triangulaire. Expliquez votre réponse.*

1. Ces graphes sont répétés ci-dessous pour vous permettre de faire des tests.

4 Problème de rendu de monnaie

On considère un système monétaire, c'est-à-dire un ensemble de **types** (de valeurs) de pièces (par exemple : des pièces de $1, 2, 5, 10, 20$ et 50 euros). Vous disposez d'autant de pièces de chaque type que nécessaire. Vous êtes vendeur et devez rendre un montant M à un client. Comment le faire en utilisant le moins de pièces possible ?

Par exemple, pour rendre 79 euros, vous pouvez rendre 79 pièces de 1 euro, ou 7 pièces de 10 euros et 9 pièces de 1 euro ou une pièce de 50 et une pièce de 20 et une pièce de 5 et deux pièces de 2. La dernière solution qui n'utilise que 5 pièces est meilleure que les deux autres.

Plus généralement, le problème prend en entrée un système $\mathcal{S} = (1 = t_1, t_2, \cdots, t_r)$ avec r types de pièces ($1 = t_1 \leq t_2 \leq \cdots \leq t_r$) et un montant $M \in \mathbb{N}$. Une solution est un r-uplet $(a_1, \cdots, a_r)$ tel que $\sum_{1 \leq i \leq r} a_i t_i = M$ et $\sum_{1 \leq i \leq r} a_i$ est minimum.

Question 9 *Prouvez que le problème admet toujours une solution.*

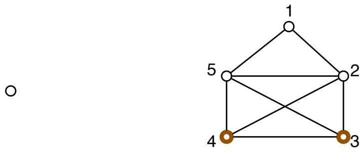
Un algorithme glouton possible est, à chaque étape, de rendre la plus grande pièce possible. Dans l'exemple, pour rendre $M = 79$, on donne d'abord 50 euros, il reste alors 29 euros à rendre, donc on utilise la plus grande pièce possible celle de 20 euros, il reste 9 euros à rendre...

Question 10 *En considérant l'exemple $\mathcal{S} = (t_1 = 1, t_2 = 3, t_3 = 4)$ et $M = 6$, prouvez que l'algorithme glouton n'est pas toujours optimal.*

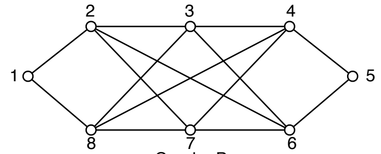
Étant donné un système $\mathcal{S} = (1 = t_1, t_2, \cdots, t_r)$, on note $OPT(M)$ une solution optimale pour un montant $M \in \mathbb{N}$. Ainsi $OPT(0) = (a_1 = 0, a_2 = 0, \cdots, a_r = 0)$.

Question 11 *Pour $M > 0$, exprimez $OPT(M)$ en fonction des $OPT(M - t_i)$ pour tout $1 \leq i \leq r$ tel que $t_i \leq M$.*

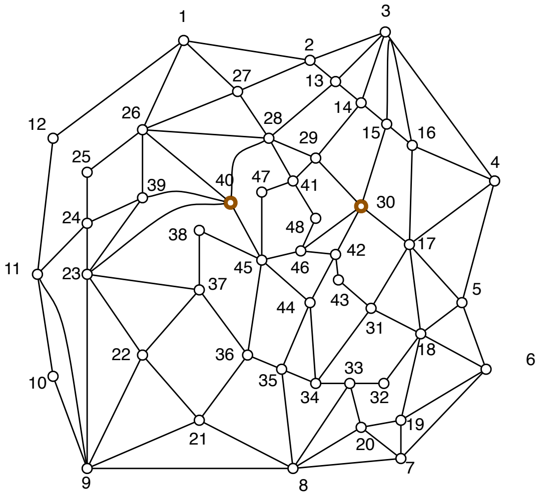
Question 12 *Ecrire un algorithme qui calcule $OPT(M)$ par programmation dynamique. Quelle est sa complexité temporelle ?*



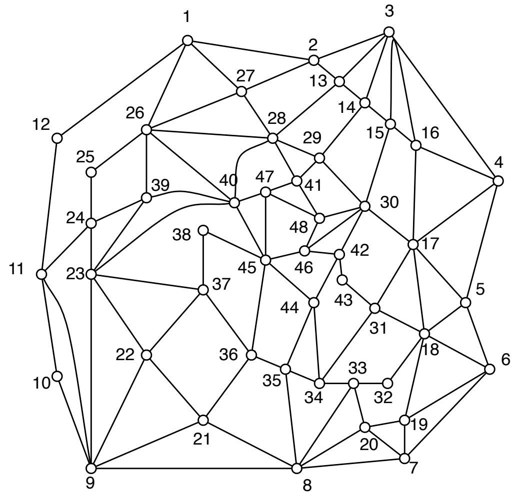
Graphe A



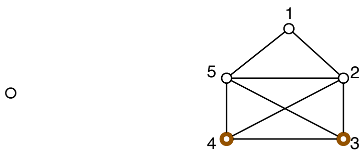
Graphe B



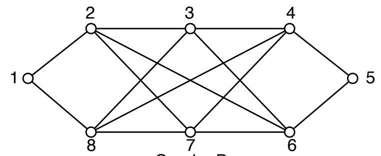
Graphe C



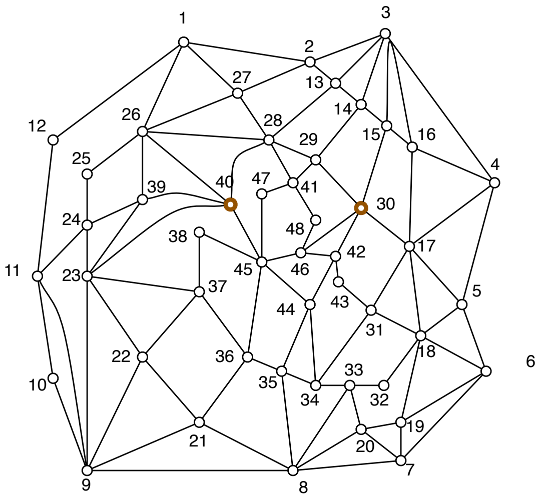
Graphe D



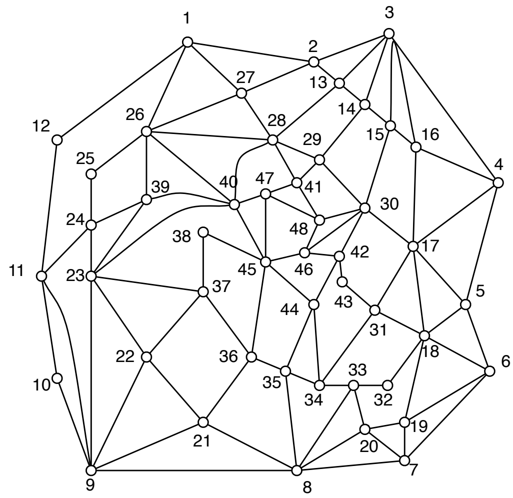
Graphe A



Graphe B



Graphe C



Graphe D