

## - TD 2. Arbres de poids minimum. -

- **Exercice 1 - Hypercube.** Soit  $n$  un entier positif non nul. L'*hypercube*  $Q_n$  est le graphe dont l'ensemble des sommets est l'ensemble des  $n$ -uplets  $(x_1, \dots, x_n)$  de 0 et de 1, deux  $n$ -uplets étant adjacents s'ils diffèrent sur une seule entrée.

- a. Dessiner  $Q_i$ , pour  $i = 1, \dots, 4$ .
- b. Déterminer le nombre de sommets et d'arêtes de  $Q_i$ .
- c. On considère la fonction de poids  $p$  définie sur les arêtes de  $Q_n$  qui vérifie  $p(e) = i$  lorsque  $i$  est l'indice de  $x_i$ , la coordonnée qui diffère entre les extrémités de  $e$ . Déterminer un arbre couvrant de poids minimum de  $Q_3$ .

- **Exercice 2 - Voyageur de commerce.** On se donne un ensemble de villes  $V = \{v_1, \dots, v_8\}$  ainsi que les distances  $d(v_i, v_j)$  séparant ces villes données par la matrice (symétrique) suivante:

$$(d(v_i, v_j)) = \begin{pmatrix} 0 & 7 & 5 & 11 & 9 & 6 & 11 & 9 \\ \cdot & 0 & 8 & 6 & 7 & 11 & 11 & 8 \\ \cdot & \cdot & 0 & 6 & 4 & 10 & 12 & 10 \\ \cdot & \cdot & \cdot & 0 & 7 & 5 & 12 & 9 \\ \cdot & \cdot & \cdot & \cdot & 0 & 7 & 12 & 10 \\ \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 11 & 9 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 10 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \end{pmatrix}$$

A un ensemble  $E$  d'arêtes sur  $V$ , on associe  $d(E)$ , la *distance totale* de  $E$ , définie comme la somme des distances des arêtes de  $E$ . Ainsi la distance totale d'un cycle  $(x_1, x_2, \dots, x_n)$  est  $d(x_1, x_2) + d(x_2, x_3) + \dots + d(x_{n-1}, x_n) + d(x_n, x_1)$ .

- a. Construire un arbre couvrant  $T$  sur  $V$  minimisant  $d(T)$ .
- b. A chaque arbre  $A$ , on associe un cycle (non nécessairement unique) comme suit:
  - Si  $A$  contient un seul sommet  $v$ , le cycle associé est  $(v)$ .
  - Si  $A$  possède une feuille  $y$ , en notant  $x$  l'unique voisin de  $y$ , un cycle associé à  $A$  est obtenu en insérant  $y$  juste après  $x$  dans un cycle associé à  $A \setminus y$ . Ainsi, lorsque  $(\dots, x, z, \dots)$  est un cycle associé à  $A \setminus y$ ,  $(\dots, x, y, z, \dots)$  est un cycle associé à  $A$ .
 Calculer un cycle associé à l'arbre  $T$  trouvé en a)
- c. On rappelle que la distance vérifie l'inégalité triangulaire  $d(a, b) \leq d(a, c) + d(c, b)$  pour tout choix de  $a, b, c$ . Montrer que si  $C$  est un cycle associé à un arbre  $A$  sur  $V$ , on a  $d(C) \leq 2d(A)$ . On pourra raisonner par induction sur le nombre  $n \geq 3$  de sommets de  $V$ .
- d. On se donne un cycle  $C'$  passant par tous les sommets de  $V$  dont la distance totale est minimum parmi les cycles couvrants. Soit  $A$  un arbre couvrant sur  $V$  de distance totale minimum parmi les arbres couvrants. Montrer que si  $C$  est un cycle associé à  $A$ , on a  $d(C) \leq 2d(C')$ .

- **Exercice 3 - Unicité.** Soit  $G = (V, E)$  un graphe connexe et  $\omega : E \rightarrow \mathbb{R}^+$  une fonction de poids sur les arêtes. On se donne deux arbres distincts  $T = (V, A)$  et  $T' = (V, A')$  de poids minimum  $\omega(A) = \omega(A')$ . On suppose que  $e$  est une arête de poids minimum parmi les arêtes de  $A \Delta A' := (A \setminus A') \cup (A' \setminus A)$ . On suppose par exemple que  $e \in (A \setminus A')$ .

- Montrer qu'il existe une arête  $e' \in (A' \setminus A)$  telle que  $\omega(e') = \omega(e)$ .
- En déduire que si la fonction  $\omega$  est injective (i.e. toutes les arêtes ont des poids différents), alors il existe un unique arbre de poids minimum.
- Quelle est la faille dans le raisonnement suivant : Si tous les poids des arêtes sont différents, on ne peut ordonner les arêtes dans l'ordre croissant de poids que d'une seule manière, donc l'algorithme de Kruskal retourne toujours la même solution, qui est ainsi l'unique arbre de poids minimum.

- **Exercice 4** - Soit  $G = (V, E)$  un graphe connexe et  $\omega : E \rightarrow \mathbb{R}^+$  une fonction de poids. Une coupe est une partition de  $V$  en deux sous-ensembles  $(X, V \setminus X)$ . Une arête  $e \in E$  traverse la coupe  $(X, V \setminus X)$  lorsque l'une des extrémités de  $e$  se trouve dans  $X$  et l'autre dans  $V \setminus X$ .

- Illustration. Dans  $Q_3$ , quelles sont les arêtes de la coupe  $\{X, V \setminus X\}$ , où  $X$  est l'ensemble des triples dont la première coordonnée est 0.
- Soit  $T$  un arbre couvrant de poids minimum dont l'ensemble des arêtes est noté  $A$ . Montrer que pour toute coupe  $C$  et toute arête  $e \in E$  traversant  $C$ , il existe  $a \in A$  traversant  $C$  telle que  $\omega(a) \leq \omega(e)$ .
- Prouver la réciproque, c'est à dire que si l'ensemble  $A$  des arêtes d'un arbre  $T$  vérifie la propriété ci-dessus, alors  $T$  est de poids minimum.

- **Exercice 5 - Algorithme de Prim.** Soit  $G = (V, E)$  un graphe connexe et  $\omega : E \rightarrow \mathbb{R}^+$  une fonction de poids. On suppose que  $G$  est codé par listes de voisins: à chaque sommet  $v$  de  $G$  est associée la pile  $Vois(v)$  des voisins de  $v$ . L'ensemble des sommets de  $V$  est  $\{x_1, \dots, x_n\}$ . Parmi les deux algorithmes suivants, préciser ceux qui retournent toujours un arbre de poids minimum.

- Initialiser  $A$  à vide.  
Pour  $i$  de 1 à  $n - 1$ , ajouter à  $A$  l'arête de poids minimum reliant  $\{x_1, x_2, \dots, x_i\}$  à  $\{x_{i+1}, \dots, x_n\}$ .  
Retourner  $A$ .
- Initialiser  $A$  à vide.  
Pour  $i$  de 1 à  $n - 1$ , ajouter à  $A$  l'arête  $x_a x_b$  de poids minimum (avec  $a < b$ ) reliant  $\{x_1, x_2, \dots, x_i\}$  à  $\{x_{i+1}, \dots, x_n\}$ , puis échanger  $x_b$  avec  $x_{i+1}$ .  
Retourner  $A$ .
- Ecrire en détail l'algorithme précédent.
- Comparer la complexité de cet algorithme avec celle de l'algorithme de Kruskal.