I Hypercube

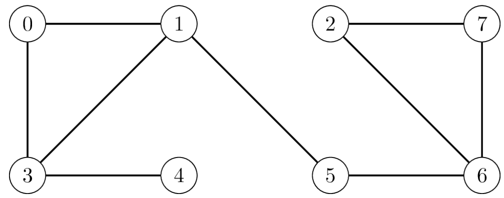
Un **hypercube** Q_n a pour sommets les mots binaires de taille n, 2 sommets étant reliés s'ils différent d'un bit.

1. Dessiner Q_3.
2. Quel est le nombre de sommets et d'arêtes de Q_n?
3. Montrer que Q_n est biparti.
4. Montrer que Q_n possède un couplage parfait.
5. Montrer que Q_n est hamiltonien : il existe un cycle (**hamiltonien**) qui visite tous les sommets exactement une fois. Dessiner un tel cycle de Q_3.

II Pont et algorithme de Tarjan

Soit $G = (V, E)$ un graphe connexe non orienté. Un **pont** de G est une arête telle que $G - e$ n'est pas connexe.

1. Donner un algorithme en complexité quadratique pour trouver tous les ponts d'un graphe G.
2. Donner tous les ponts du graphe ci-dessous.



Soit $r \in V$. Soit T une arborescence de parcours en profondeur depuis r.

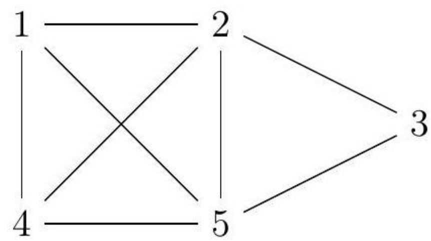
On définit un graphe orienté $\vec{G}$ en orientant les arcs de T dans le sens du parcours et les arcs arrières en direction de l'ancêtre. On définit $p(v)$ comme étant la profondeur du sommet v dans T. On définit $l(v)$ comme étant la plus petite valeur de $p(u)$ pour un sommet u accessible (par un chemin) depuis v dans $\vec{G}$.

3. On effectue un parcours en profondeur depuis 0 dans le graphe de la question 2, en visitant à chaque fois le plus petit sommet en priorité lorsqu'il y a plusieurs choix. Dessiner $\vec{G}$ et les valeurs de $p(v)$ et $l(v)$ pour chaque sommet v.
4. À quelle condition sur l un arc (u, v) de T est-il un pont ?
5. Donner une relation de récurrence permettant de calculer $l(v)$.
6. En déduire un algorithme en OCaml permettant de trouver tous les ponts d'un graphe G. On choisira des structures de données adaptées. Quelle est la complexité de cet algorithme ?

III Calcul des triangles

On considère un graphe non-orienté $G = (V, E)$ où $V = \{1, \ldots, n\}$ est l'ensemble des sommets et E est un ensemble d'arêtes qui sont des paires de sommets de V. Un triangle dans G est un sous-ensemble $\{x, y, z\} \in V$ tel que $\{x, y\}, \{y, z\}$, et $\{x, z\}$ appartiennent à E. Dans ce problème, on veut concevoir des algorithmes qui prennent en entrée un graphe et calculent (sans doublons) l'ensemble des triangles du graphe.

1. Déterminer les triangles du graphe suivant:



2. Si G est représenté par matrice d'adjacence, proposer un algorithme naïf en $O\left(|V|^3\right)$ pour déterminer l'ensemble des triangles de G.

Dans la suite du sujet, on supposera toujours que le graphe d'entrée est fourni sous forme de listes d'adjacence, et on supposera toujours que chacune de ces listes est triée.

3. Soient l_1 et l_2 deux listes triées. Proposer un algorithme en complexité linéaire pour calculer l'intersection de ces deux listes (les éléments appartenant à la fois à l_1 et l_2).
4. Soit Δ le degré maximal d'un sommet de G. Proposer un algorithme en temps $O(|E| \times \Delta)$ pour déterminer l'ensemble des triangles de G. Commenter la performance de cet algorithme.
5. Si l'on compare l'algorithme de la question 2 et celui de la question 4, lequel a la meilleure complexité? Le choix de la représentation du graphe d'entrée était-il important?
6. Un sommet lourd est un sommet de degré $\geq \sqrt{E}$. Montrer que G contient au plus $2\sqrt{E}$ sommets lourds.
7. Proposer un algorithme en $O\left(|E|^{3/2} + |V|\right)$ pour déterminer l'ensemble des triangles de G.

On suppose dans la suite que G n'a pas de triangle et on veut majorer le nombre d'arêtes de G.

8. Montrer que pour toute arête $e = \{u, v\}$ de G, $\deg(u) + \deg(v) \leq |V|$.
9. En déduire que $\displaystyle\sum_{v \in V} \deg(v)^2 \leq |V||E|$.
10. En déduire que $|E| \leq \dfrac{|V|^2}{4}$ (théorème de Mantel).
11. Donner un exemple de graphe sans cycle de longueur 3 et vérifiant $|E| = \dfrac{|V|^2}{4}$.

IV Questions sur les arbres couvrants

Soit $G = (V, E)$ un graphe pondéré.

1. Soit C un cycle de G et $e = \{u, v\}$ une arête de C dont le poids est strictement supérieur au poids des autres arêtes de C. Montrer que e ne peut pas appartenir à un arbre couvrant de de poids minimum de G.
2. (Propriété d'échange) Soient T_1, T_2 deux arbres couvrants de G et e_2 une arête de $T_2 - T_1$. Montrer qu'il existe une arête e_1 de T_1 telle que $T_1 - e_1 + e_2$ (le graphe obtenu en remplaçant e_1 par e_2 dans T_1) est un arbre couvrant de G.
3. Montrer que si tous les poids des arêtes de G sont différents, alors G admet un unique arbre couvrant de poids minimum.
4. Soit T_1 un arbre couvrant de poids minimum de G et T_2 le 2ème plus petit arbre couvrant, c'est-à-dire l'arbre couvrant de poids minimum en excluant T_1. Montrer que T_1 et T_2 diffèrent d'une arête et en déduire un algorithme pour trouver T_2.

V Questions sur les couplages

1. Soit G un graphe. Montrer que si G a un couplage parfait alors G possède un nombre pair de sommets. La réciproque est-elle vraie ?

2. Soit M_1 et M_2 deux couplages d'un graphe G, avec M_2 maximal. Montrer que $|M_1| \leq 2|M_2|$, puis donner un cas d'égalité.

3. Soit $M = (m_{i,j})$ une matrice de taille $n \times n$. On définit le permanent de M :

$$\text{per}(M) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} m_{i,\sigma(i)}$$

   où S_n est l'ensemble des permutations de $\{1, \ldots, n\}$.
   Définir un graphe G dont le nombre de couplages parfaits est égal à $\text{per}(M)$.
   Remarque : il n'existe pas d'algorithme efficace pour calculer le permanent d'une matrice, contrairement au déterminant qui peut être calculé en $O(n^3)$ avec l'algorithme du pivot de Gauss.