Feuille d'exercices n°13 - Graphes

Notions abordées

- arbre au sens de graphe connexe acyclique
- accessibilité dans un graphe implicite
- tri topologique d'un graphe orienté et détection de circuit

Exercice 1 Tri topologique

On dit qu'une liste L d'éléments $L_1, L_2, \dots L_n$ est un **tri topologique** (des sommets) d'un graphe orienté $G = (S, A)$ ssi $\{L_i \,|\, i \in [1..n]\} = S$ et $\forall (i,j) \in [1..n]^2$, $i < j \Rightarrow (L_j, L_i) \notin A$. On dit on parfois que L est une permutation des sommets dans laquelle aucun sommet n'est placé après l'un de ses prédécesseurs.

Question 1
Donner un exemple de tri topologique d'un graphe orienté qui n'en est pas un parcours.

Question 2
Donner un exemple de parcours d'un graphe orienté qui n'en est pas un tri topologique.

Question 3
Que peut-on dire du premier sommet d'un tri topologique. Combien a-t-il de prédécesseurs ? de successeurs ? Et le dernier ?

Question 4
Existe-t-il toujours un tri topologique ? Donner une condition nécéssaire et suffisante à l'existence d'un tri topologique dans un graphe.

Question 5
Proposer un algorithme en pseudo-code qui permet de calculer un tri topologique s'il en existe un, et qui donne une preuve (au sens témoin, pas au sens démonstration) qu'il n'en existe pas sinon. *On peut essayer de construire le tri topologique pas à pas en remarquant que si $(L_i)_{i\in[1..n]}$ est un tri topologique de $G = (S, A)$, alors $(L_i)_{i\in[2..n]}$ est un tri topologique du graphe induit par $S \setminus \{L_1\}$.*

Question 6
En utilisant la représentation qui permet la complexité la plus intéressante, implémenter l'algorithme proposé. *On attend une complexité pire cas en $O(n^2)$.*

Question 7
Comment détecter si un graphe orienté présente des circuits.

Exercice 2 Définitions équivalentes de la notion d'arbre ✍

Soit $G = (S, A)$ un graphe **non orienté**. On note $n = |S|$ et $m = |A|$. De plus on note K_n le graphe complet sur S, c'est-à-dire $K_n = \left(S, \left\{ \{u, v\} \,|\, (u, v) \in S^2 \text{ avec } u \neq v \right\}\right)$. On rappelle qu'un sous-graphe de K_n est de la forme (S, B) avec $B \subseteq A$. On peut alors ordonner les sous-graphes de K_n par l'inclusion de leur ensemble d'arrêtes, *i.e.* $(S, B) \preccurlyeq (S, C)$ ssi $B \subseteq C$.

Montrer que les 5 propositions ci-dessous sont équivalentes.

- G est connexe et acyclique
- G est connexe et $|A| = |S| - 1$
- G est acyclique et $|A| = |S| - 1$
- G est maximal parmi les sous-graphes connexes de K_n
- G est minimal parmi les sous-graphes acycliques de K_n

Exercice 3 Exploration du graphe $\mathfrak{S}_n$

On s'intéresse à une étude empirique des sous-ensembles de $\mathfrak{S}_n$ générant $\mathfrak{S}_n$, c'est-à-dire des ensembles tel que le plus petit sous-groupe de $\mathfrak{S}_n$ qui les contient est $\mathfrak{S}_n$ lui-même.

Une permutation de $\mathfrak{S}_n$ de la forme $\begin{pmatrix} 0 & 1 & 2 & \cdots & n-2 & n-1 \\ a_0 & a_1 & a_2 & \cdots & a_{n-2} & a_{n-1} \end{pmatrix}$ sera représentée par un tableau OCaml `[|`a_0`;` a_1`;` a_2`; ...;` a_{n-2}`;` a_{n-1}`|]`.[1]
On suppose donc défini le type suivant : `type permutation = int array`.

Question 1
Écrire une fonction `id_n` prenant en argument un entier naturel `n` et renvoyant $id_{[0..n[}$, c'est-à-dire la permutation *i.e.* $\begin{pmatrix} 0 & 1 & 2 & \cdots & n-2 & n-1 \\ 0 & 1 & 2 & \cdots & n-2 & n-1 \end{pmatrix}$.

Question 2
Écrire une fonction `inverse : permutation -> permutation` prenant en argument une permutation `p` et renvoyant la permutation inverse $\texttt{p}^{-1}$.

Question 3
Écrire une fonction `compose : permutation -> permutation -> permutation` prenant en arguments deux permutations `p1` et `p2`, et calculant la permutation composée `p1` $\circ$ `p2`.

Étant donné un sous-ensemble $X \subseteq \mathfrak{S}_n$, on considère alors la relation R sur $\mathfrak{S}_n$ définie par xRy ssi $\exists z \in X, x = z \circ y$ ou $x = z^{-1} \circ y$ ou $x = y \circ z$ ou $x = y \circ z$. On remarque alors que le sous-groupe de $\mathfrak{S}_n$ engendré par X est la composante connexe de $(\mathfrak{S}_n, R)$ contenant l'identité.

Question 4
Au moyen d'un parcours de graphe, définir une fonction `card_groupe_engendre` prenant en argument une liste de permutations et calculant le cardinal du sous-groupe de $\mathfrak{S}_n$ engendré par les permutations de cette liste. *On ne construira pas en mémoire le graphe $(\mathfrak{S}_n, R)$, on se contentera de le parcourir.*

Question 5
En déduire une fonction `est_generateur` décidant si une liste de permutations de $\mathfrak{S}_n$ engendre $\mathfrak{S}_n$.

[1] Noter l'indexation à partir de 0.