- Examen. Durée 2h. -

- Exercice 1 - Arbres aléatoires.

Soit A un arbre avec racine dont les sommets sont étiquetés par des entiers distincts. On associe à A un mot $L(A)$ sur l'alphabet X de la manière suivante :

- si A n'a qu'un seul sommet (racine), $L(A)$ est le mot vide.

- sinon, on trie dans l'ordre croissant les feuilles $f_1, \dots, f_k$ de A et on forme le mot $L' := p(f_1)p(f_2)\dots p(f_k)$ où $p(f_i)$ est le pêre de f_i. En notant A' l'arbre obtenu en supprimant les feuilles de A, on pose $L(A) = L'.L(A')$ où . est la concaténation. Ainsi l'arbre A sur $\{1, \dots, 4\}$ tel que 1 est racine et $p(2) = 1$, $p(3) = 1$, $p(4) = 3$ est codé par $L(A) = 131$

a. Calculer $L(A)$ lorsque A est l'arbre sur $\{1, 2, \dots, 9\}$ dont la racine est 8, et dont les arêtes sont $\{48, 38, 14, 24, 64, 93, 51, 71\}$.

b. Quel arbre A sur $\{1, 2, \dots, 9\}$ est codé par $9, 9, 1, 9, 9, 7, 2, 7$?

c. Peut-on utiliser ce codage afin d'effectuer un tirage aléatoire uniforme d'arbres enracinés ?

d. En déduire qu'il existe n^{n-2} arbres sur $\{1, \dots, n\}$.

e. Illustrer cette formule pour $n = 4$.

- Exercice 2 - Automates.

Soit L le langage sur l'alphabet $\{0, 1\}$ dont les mots sont les écritures en binaire des nombres qui sont des multiples de 3.

a. Proposer un automate fini déterministe A qui reconnait L. On adoptera ici un changement de convention en supposant que L est lu de la droite vers la gauche. (On remarquera pour cela que 2^n est égal à 1 modulo 3 si n est pair et à 2 modulo 3 sinon.)

b. Proposer un automate fini non déterministe B qui reconnait L, cette fois-ci en supposant que L est lu de la gauche vers la droite.

c. Montrer qu'il n'existe pas d'automate fini reconnaissant les écritures décimales des carrés.

- Exercice 3 - Coupe minimum.

Soit $G = (V, E)$ un graphe orienté et x, y deux sommets de V. Une *xy-coupe* est une partition X, Y de V telle que $x \in X$ et $y \in Y$. La *valeur* d'une coupe X, Y est le nombre d'arcs orientés uv tels que $u \in X$ et $v \in Y$.

a. Montrer que s'il existe une xy-coupe de valeur 0 alors il n'existe pas de chemin orienté de x à y.

b. Montrer la réciproque en exhibant un xy-coupe de valeur 0 lorsqu'il n'existe pas de chemin orienté de x à y.

c. Soit P un chemin orienté de x à y. Montrer que si on inverse l'orientation de tous les arcs de P, alors la valeur de toute xy-coupe diminue exactement de 1.

d. Décrire un algorithme polynomial calculant une xy-coupe de valeur minimum.

e. Proposer un algorithme déterministe qui calcule une coupe minimum dans un graphe non orienté. Comparer sa complexité à celle de l'algorithme randomisé de Karger.

- Exercice 4 - Programmation dynamique.

Dans cet exercice, on suppose que $k \geq 64$.

a. Montrer que si $S = \{x_1, \ldots, x_k\}$ est un ensemble d'entiers tel que tous les x_i ont valeur au plus k^9, alors il existe deux sous-ensembles non vides disjoints S_1 et S_2 de S qui ont même somme.

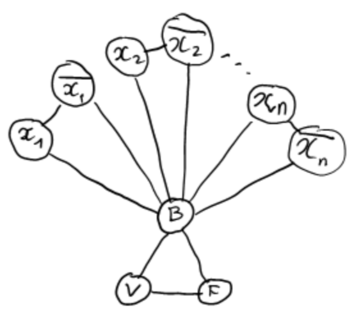
b. Proposer un algorithme polynomial en k qui sous les hypothèses précédentes trouve effectivement deux tels ensembles S_1 et S_2.

- Exercice 5 - NP-complétude.

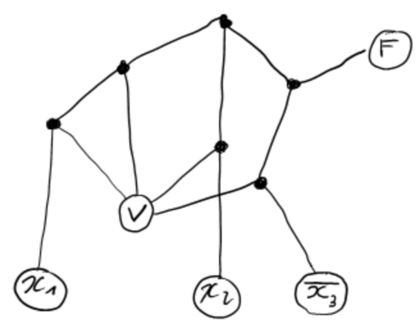
Une graphe $G = (V, E)$ est *3-colorable* s'il existe une application c de V dans $\{1, 2, 3\}$ telle que pour toute arête $xy \in E$, on a $c(x) \neq c(y)$.

a. Montrer que le problème de décision : G est-il 3-colorable ? est dans NP.

b. Montrer que ce problème est en fait NP-complet. Préciser comment utiliser les graphes ci-dessous afin de réduire 3-SAT à 3-COLORATION :



Variables



Clause $x_1 \vee x_2 \vee \overline{x_3}$