TD 6

Exercice 1. *Nombres premiers*

1. Soit n et N deux entiers ($n < N$). Si on tire un nombre premier aléatoire inférieur ou égal à n, quelle est la probabilité qu'il divise N ?
2. Comment choisir n pour que la probabilité précédente soit supérieure ou égale à $1/2$.

Exercice 2. *Isomorphisme d'arbres*

Un arbre *enraciné* est un arbre dont on fixe la racine. On le représente par un triplet (V, E, r) où V est l'ensemble des sommets, $E \subset V \times V$ l'ensemble des arêtes, et $r \in V$ la racine. On dit que deux arbres (enracinés) $A_1 = (V_1, E_1, r_1)$ et $A_2 = (V_2, E_2, r_2)$ sont *isomorphes* s'il existe une bijection $f : V_1 \to V_2$ telle que $f(r_1) = r_2$ et vérifiant la propriété suivante pour tout $v \in V_1$: si les fils de v sont v_1, …, v_k, alors les fils de $f(v)$ sont $f(v_1)$, …, $f(v_k)$. Aucun ordre n'est imposé sur les fils d'un nœud.

1. Montrer que si A_1 et A_2 sont isomorphes, alors pour tout sommet v de A_1, le sous-arbre de A_1 enraciné en v est isomorphe au sous-arbre de A_2 enraciné en $f(v)$.

La hauteur d'un nœud v dans un arbre est défini récursivement : une feuille est de hauteur 0, et un sommet v est de hauteur h si la hauteur maximale de ses fils est $h - 1$.
À tout sommet v d'un arbre A, on associe un polynôme P_v que l'on définit récursivement. Si v est une feuille, $P_v = x_0$. Si v a pour fils v_1, …, v_k, et que v est à hauteur h, alors

$$P_v = \prod_{i=1}^{k} (x_h - P_{v_i})$$

où pour tout i, P_{v_i} est le polynôme associé à v_i.

2. Montrer par induction que A_1 et A_2 sont isomorphes si et seulement si $P_{r_1} = P_{r_2}$ où r_1 et r_2 sont les racines respectives de A_1 et A_2.
3. En déduire un algorithme probabiliste de complexité linéaire qui détermine si deux arbres sont isomorphes.

Exercice 3. *Taille des listes à raccourcis*

✎ Une liste à raccourcis utilise de la redondance dans la représentation des données pour accélérer les opérations à effectuer. Montrer que cette redondance reste raisonnable : l'espérance de la taille d'une liste à raccourcis de n éléments est $O(n)$.

Exercice 4. *Filtres de Bloom*

On s'intéresse dans cet exercice à une structure de données qui permet de stocker de manière très compressée un ensemble (statique, c'est-à-dire duquel on ne supprime jamais d'élément). La contrepartie est la présence de faux-positifs : notre structure de données répond que x appartient à l'ensemble alors que ça n'est pas le cas.
Un filtre de Bloom pour un ensemble de taille n est donné par un entier m (la taille de la représentation) et k fonctions de hachage h_1, …, h_k indépendantes. L'ensemble X est représenté par un mot booléen w de taille m : pour chaque élément $x \in X$, on met les k bits de w correspondants ($w_{h_1(x)}$, …, $w_{h_k(x)}$) à 1. Un bit peut être mis plusieurs fois à 1.
Dans la suite, on suppose qu'on a construit la représentation w d'un ensemble X de taille n. On se place dans le modèle aléatoire pour les fonctions de hachage.

1. Pour chaque i, quelle est la probabilité p que le i-ème bit de w soit égal à 0 ?

Dans la suite, on fait l'hypothèse simplificatrice suivante : on suppose qu'une fraction p des bits de w sont à 0 [1].

2. Quelle est la probabilité q d'avoir un faux-positif (sous l'hypothèse simplificatrice) ?

3. Montrer qu'en prenant $k = m \ln 2/n$, on obtient une probabilité de faux-positif exponentiellement petite. *Indication : on pourra utiliser que pour tout* $m > 1$, $(1 - 1/m)^m \leq 1/e$.

1. Pourquoi ça n'est pas une conséquence de la question précédente ?