AAGA Examen Réparti 1

Appareils électroniques interdits. Seuls documents autorisés : Notes de cours et de TD/TME.
Le barème est indicatif. L'examen est noté sur 20, il y a donc 1 points en bonus.

Exercice* 1 *: Générateur congruentiel quadratique (3 points)

On définit un *générateur congruentiel quadratique*, un générateur du type :

$$X_{n+1} = X_n \times (X_n + 1) \mod 16, \qquad \text{pour } n \geq 0.$$

Le générateur a besoin d'une graine X_0 pour être bien défini.

On rappelle les 3 égalités suivantes :
$12 \times 13 \mod 16 = 12 \qquad 13 \times 14 \mod 16 = 6 \qquad 14 \times 15 \mod 16 = 2.$

Question 1 Étant donné une graine $X_0 \in \mathbb{N}$ quelconque, rappeler la définition de la période du générateur pour cette graine. Prouver que la période maximale de ce générateur est 16, même si $X_0 \geq 16$.

Question 2 Soit $X_0 = 1$. Donner la période issue de X_0 : sa longueur et la suite d'entiers qui se répète.

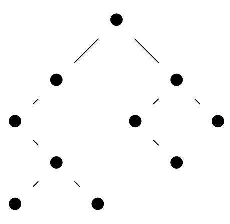
Question 3 Soit $X_0 = 14$. Donner la période issue de X_0 : sa longueur et la suite d'entiers qui se répète.

Question 4 Rappeler la définition de la période d'un générateur. Quelle est la période de ce générateur ?

Exercice* 2 *: Générateur d'arbres binaires incomplets (9 points)

Un arbre binaire incomplet est un arbre dont les nœuds sont dans une des quatre configurations suivantes. Chaque nœud possède soit :
- 0 enfant
- 1 enfant gauche qui est un arbre binaire incomplet
- 1 enfant droit qui est un arbre binaire incomplet
- 2 enfants qui sont chacun des arbres binaires incomplets.

La taille d'un arbre binaire incomplet est égale au nombre de nœuds qu'il contient. Par la suite, le terme *arbre* aura pour signification arbre binaire incomplet.
Voilà un exemple d'arbre de taille 10.



Question 1 Dessiner tous les arbres de taille $1, 2, 3$ et 4.

Question 2 On note B_n le nombre d'arbres de taille n. Donner la formule de récurrence vérifiée par B_n. Ne pas oublier le cas de base.

Question 3 Calculer le nombre d'arbres de taille 5.

Générateur d'arbre à la Rémy

Voilà un algorithme itératif permettant de construire un arbre de taille $n \geq 1$:

- On initialise l'arbre à l'unique arbre de taille 1 : il contient une feuille, et on l'étiquette par 1.
- Puis on répète cette étape tant que l'arbre n'a pas la taille n.
  A l'étape i, on part d'un arbre A de taille i. On choisit parmi ses i nœuds un seul nœud, avec la distribution non uniforme décrite ci-dessous. Chaque feuille de l'arbre A a pour poids 3. Chaque nœud unaire (avec 1 enfant gauche ou 1 enfant droit) a pour poids 2 et chaque nœud binaire a pour poids 1. Ceci signifie par conséquent, qu'une feuille a trois fois plus de chance d'être choisie qu'un nœud binaire.

  Si on a choisi une feuille, alors de façon équiprobable, on lui greffe un enfant gauche (qui est une feuille) ou on lui greffe un enfant droit (qui est une feuille), ou on lui ajoute un nœud au dessus-d'elle : entre elle-même et son parent [1] (dans ce cas, il faut en plus tirer à pile ou face pour choisir si l'ancienne feuille choisie devient enfant gauche ou enfant droit de ce nouveau nœud parent).
  Si on a choisi un nœud unaire, alors de façon équiprobable, on lui greffe un deuxième enfant (du côté vide et qui est une feuille) ou on lui ajoute un nœud au dessus-de lui : entre lui-même et son parent [1] (dans ce cas, il faut en plus tirer à pile ou face pour choisir si l'ancien nœud choisi devient enfant gauche ou enfant droit de ce nouveau nœud parent).
  Si on a choisi un nœud binaire, alors on lui ajoute un nœud au dessus-de lui : entre lui-même et son parent [1] (dans ce cas, il faut en plus tirer à pile ou face pour choisir si l'ancien nœud choisi devient enfant gauche ou enfant droit de ce nouveau nœud parent).
  Le nouveau nœud ajouté prend l'étiquette $i + 1$
- On renvoie l'arbre de taille n sans étiquette (dans les nœuds).

Question 4 Construire via ce processus un arbre de taille 5, en donnant à chaque étape les valeurs des entiers aléatoires nécessaires lors de la construction de cet exemple.

Question 5 Donner une structure de données permettant de représenter ces arbres.

Question 6 Donner le pseudo-code de la fonction de génération d'arbre prenant un unique paramètre n, la taille de l'arbre à générer. On fera appel à une fonction *RandInt* prenant un entier p en paramètre en renvoyant un entier uniformément dans l'intervalle $\{0, 1, 2, \ldots, p - 1\}$. Faire attention à l'efficacité aussi bien en complexité temporelle et spatiale, que par rapport au nombre de bits aléatoires nécessaires aux différents appels à *RandInt*.

Question 7 Justifier les trois complexités mentionnées à la question précédente.

Question 8 Si ce n'est pas le cas, peut-on améliorer l'algorithme pour obtenir une complexité linéaire en n ?

1. s'il n'y a pas de parent, le nouveau nœud devient racine de l'arbre.

Veuillez utiliser une nouvelle copie pour les deux exercices suivants.

Exercice* 3 *: Coefficient de clustering dans les graphes de Erdös-Rényi sous le modèle* $\mathbb{G}_{n,p}$ *(3 points)

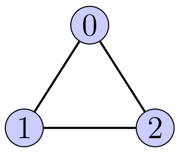
Nous cherchons à estimer le coefficient de clustering quand $n \to \infty$. Pour cela nous utiliserons des résultats vus en cours que l'on combinera ensemble. Nous utiliserons le résultat suivant sur le nombre moyen d'apparition d'un **motif** m avec k nœuds et ℓ arêtes.

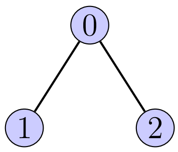
$$\bar{n}_F(m) = \mathbb{E}[n_F] \overset{n\to\infty}{=} \binom{n}{k} \frac{k!}{a_F(m)} p^\ell \approx \frac{n^k p^\ell}{a_F(m)},$$

où n est le nombre de nœuds du graphe et $a_F(m)$ est le nombre d'automorphismes du motif m.

Remarque : Rappelez-vous que k est le nombre de nœuds du motif m qui lui est fixe alors que n est le nombre de nœuds du graphe qui tend à l'infini.

Soient les motifs suivants nommés respectivement triangle et triade :





Question 1 Donner le nombre d'automorphismes de chacun de ces deux motifs.

Rappel : Automorphisme :
Un automorphisme $f : V \to V$ d'un graphe $G = (V, E)$ est une permutation des nœuds du graphe telle que :

$$\forall (u, v) \in V^2, ((u, v) \in E \Leftrightarrow (f(u), f(v)) \in E)$$

Question 2 Appliquer la formule de $\bar{n}_F(m)$ pour chacun des deux motifs et donner son approximation.

À présent, nous supposons que $p = \dfrac{c}{n}$ pour une constante positive c.

Question 3 Rappeler la formule du coefficient de clustering global d'un graphe et appliquer-là en remplaçant le nombre de triangles et triades par leur valeurs moyenne trouvées dans la question précédente et en remplaçant p par sa valeur ci-dessus.

Question 4 Que trouve-t-on quand $n \to \infty$? Que pouvez-vous déduire du coefficient de clustering global d'un graphe dans le modèle $\mathbb{G}_{n,p}$?

Exercice* 4 *: Implémentation de l'algorithme de Barabási–Albert (BA) (6 points)

Nous rappelons le principe de construction d'un graphe de BA :
Étant donné $n_0, m \in \mathbb{N}^*$ représentant respectivement le nombre de nœuds du graphe de départ et le nombre de nouvelles arêtes ajoutées à chaque instant de temps. Ces entiers sont tel que $m \leq n_0$. Nous dénotons aussi par n_t, ℓ_t respectivement le nombre de nœuds au temps t et le nombre d'arêtes au temps t.

1. Au temps 0 on commence avec un graphe complet sur n_0 nœuds (les nœuds sont étiquetés $0, 1, \ldots, n_0 - 1$).
2. Pour $t > 1$, un nouveau nœud est ajouté au graphe portant l'étiquette $n = n_0 + (t-1)$, et on ajoute m arêtes au graphe à partir du nouveau nœud n vers des nœuds déjà existants suivant la probabilité suivante :
$$P(n \to i) = \frac{d_{i,t-1}}{\sum_j d_{j,t-1}} = \frac{d_{i,t-1}}{2\ell_{t-1}},$$
où $P(n \to i)$ est la probabilité que le nœud n se rattache au nœud i et $d_{i,k}$ est le degré du nœud i au temps k.

Question 1 Donner la formule du nombre d'arêtes que contient le graphe complet en fonction de son nombre de nœuds n_0 ? Justifier votre réponse.

Question 2 Donner la formule du nombre de nœuds n_t en fonction de n_0 et t. Donner aussi la formule de ℓ_t en fonction de n_0, m et t. Justifier vos réponses.

Question 3 Supposons que l'on itère la procédure pour un nombre de temps égal à $N - n_0$. Soit K le nombre d'arêtes dans le graphe final. Donner la formule de K en fonction de m, n_0 et N. Justifier votre réponse.

Remarque : Vous n'avez pas besoin d'avoir répondu aux questions précédentes pour faire le reste de l'exercice.

Nous voulons maintenant écrire un pseudo-code efficace pour engendrer un graphe de BA de paramètres n_0 et m et ayant N nœuds au final. Pour ce faire nous allons nous aider de deux vecteurs $\mathbf{i}$ et $\mathbf{j}$ de taille K où K (où K est comme dans la question précédente le nombre d'arêtes dans le graphe final) qui pourront contenir des entiers (entre 0 et $N-1$) représentant des nœuds du graphe. À un indice donné r les valeurs $\mathbf{i}[r], \mathbf{j}[r]$ représente une arête entre les nœuds $\mathbf{i}[r]$ et $\mathbf{j}[r]$ du graphe.

Exemple : Sur le graphe triangle de l'exercice précédent nous pouvons décrire les arêtes avec

$$\mathbf{i} = (0, 0, 1), \quad \mathbf{j} = (1, 2, 2)$$

Question 4 Écrire le pseudo-code d'une fonction GRAPHE_COMPLET($\mathbf{i}, \mathbf{j}, n_0$) remplissant les vecteurs $\mathbf{i}, \mathbf{j}$ pour qu'ils représentent un graphe complet sur les nœuds d'indices allant de 0 à $n_0 - 1$ et donner sa complexité.

Question 5 Écrire le pseudo-code de la fonction BA(N, n_0, m) renvoyant un graphe de BA contenant N nœuds de paramètres n_0 et m. Dans votre pseudo-code vous pourrez vous servir des fonctions :

- VECTEUR(N,n_0,M) qui renvoie un vecteur de taille 'nombre d'arêtes dans le graphe final' où toutes les cases sont initialisées à *nil*.
- GRAPHE_COMPLET($\mathbf{i}, \mathbf{j}, n_0$)
- NŒUD_ALEATOIRE($\mathbf{i}, \mathbf{j}, \ell, n$) où ℓ est le nombre d'arêtes actuel dans le graphe et n l'étiquette du nouveau nœud. Cette fonction renvoie l'étiquette d'un nœud ν selon la distribution des $P(n \to i)$ et s'assure que n et ν ne sont pas reliés par une arête.

Question 6 En supposant que NŒUD_ALEATOIRE($\mathbf{i}, \mathbf{j}, \ell, n$) s'exécute en $O(m)$ et VECTEUR(N,n_0,M) en $O(n_0^2)$. Quelle est la complexité de BA(N, n_0, m) ?