Université de Paris

Partiel d'Algorithmique

Durée 2 heures

Important : Tout document du cours/TDs autorisé. Pas de photocopies de livres. Internet et téléphones sont strictement interdits (!). Il sera tenu compte de la clarté de vos explications et illustrations. Bien numétoter les exercices et les questions.

Exercice 1 : Algorithme randomizé et structures aléatoires

Dans cet exercice, on considère l'algorithme suivant qui va générer un graphe aléatoire avec n sommets (étiquettes) et M arêtes.

Algorithme 1 : RandomGraphGenerator

Data : n, M
Result : Un graphe aléatoire $\mathbb{G}(n, M)$
$V := \{1, \cdots, n\}$;
$E := \emptyset$;
($\star$ $G = (V, E) =$ graphe vide $\star$)
$r := 0$;
for $i := 1$ *to* M (*$\star$ i est le compteur d'arêtes $\star$*)
do
  repeat
    Prendre x uniformément au hasard (u.a.r.) dans l'ensemble V.
    Prendre y u.a.r. dans l'ensemble $V - \{x\}$.
    $r := r + 1$; ($\star$ r est un *compteur* de **répétitions** sur les (x, y) déjà liés $\star$)
  until $(x, y) \notin E$
  (*$\star$ Répéter jusqu'à trouver* **deux sommets distincts non encore reliés** *par une arête $\star$*)
  $E := E \bigcup \{(x, y)\}$; ($\star$ on rajoute l'arête (x, y) dans G $\star$)
  $i := i + 1$;
end
return (V, E) ; ($\star$ on renvoie le graphe $G = (V, E)$ $\star$)

1. Dans cette question, on supposera que $n^{7/8}$ est un entier. Montrer que si $M = n^{7/8}$ alors le compteur r vérifie $r = M$ avec une probabilité tendant vers 1 avec n tendant vers l'infini ("vraisemblablement, il n'y aura pas de retirages/répétitions").
2. Dans cette question, on supposera que $n^{1/4}$ est un entier. Montrer que si $M = n^{1/4}$ alors l'algorithme RandomGraphGenerator(n, M) ne produit qu'un graphe aléatoire avec des *sommets isolés* et des *arêtes isolées* avec une probabilité tendant vers 1 quand $n \to +\infty$.
3. Pour quelles valeurs du paramètre M, r atteint $r = n(n-1)/2$ avec probabilité presque 1 ?

Exercice 2 : Algorithmique randomizée

Dans cet exercice on s'intéresse aux problèmes d'élection (vus en cours) dans les modèles synchrones quand le graphe support du réseau est complet et anonyme (les processeurs sont non identifiés).

1. Dans cette question, on considère le modèle de "radio networks" avec exactement 2 processeurs communiquant via un canal radio (avec ou sans détecteur de collision). A un instant discret t, chacun des deux processeurs tente d'accaparer le canal avec probabilité p. Calculer la probabilité qu'un des deux processeurs réussisse à obtenir seul le canal.

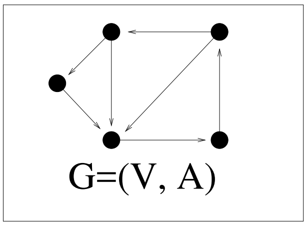
2. On se fixe $\varepsilon > 0$. On dira dans cette partie qu'un algorithme d'élection réussit avec très forte probabilité s'il succède à élire un leader avec une probabilité d'au moins $1 - O(\varepsilon)$. Dans les mêmes hypothèses que la question précédente, montrer qu'on ne peut pas élire un leader avec très forte probabilité en moins de $\Omega(\log \frac{1}{\varepsilon})$ * .

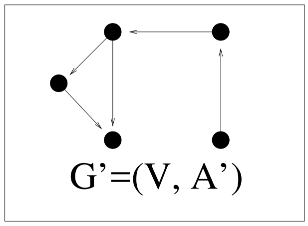
3. Généraliser les deux questions précédentes dans le cas d'un réseau complet à k nœuds, i.e. (i) calculer la probabilité qu'une élection ait lieu à un temps quelconque dans un réseau à k processeurs et (ii) montrer qu'il faut un temps logarithmique pour avoir un tel succès avec très forte probabilité.

4. Déduire des questions précédentes un algorithme Monte Carlo pour élire un leader parmi n processeurs dans le cas d'un réseau avec detecteur de collision. Votre algorithme doit s'arrêter en un temps connu d'avance avec un taux d'erreur de moins de $1/n^3$.

Exercice 3 : Problèmes d'approximation dans les graphes dirigés

Dans cet exercice on s'intéresse aux graphes dirigés (ou digraphes) et en particulier aux sous-graphes acycliques. Le problème algorithmique est le suivant.
Input : Un graphe dirigé $G = (V, A)$.
Output : $A' \subseteq A$ tel que le digraphe $G' = (V, A')$ ne contienne pas de cycle orienté.
Optimisation : *maximiser* $|A'|$, i.e. le nombre d'arcs dans G'.





1. En observant qu'un sommet s appartenant à un cycle orienté possède toujours un arc rentrant et un arc sortant proposer un algorithme glouton offrant une solution au problème ci-dessus (sans l'optimisation).

2. Quel est le facteur d'approximation de votre algorithme ?

Exercice 4 : Formules CNF (forme normale conjonctive ou Conjunctive Normal Form)

Une formule CNF $F_{n,m}$ est donnée par $C_1 \wedge C_2 \wedge \cdots \wedge C_m$ avec chaque clause C_i est de la forme $y_1 \vee y_2 \vee \cdots y_k$ avec les $y_i \in \{x_1, x_2, \cdots, x_n, \bar{x}_1, \bar{x}_2, \cdots, \bar{x}_n\}$ et k est un entier positif quelconque. On dit qu'une instance $F_{n,m}$ est SAT s'il existe une affectation des variables booléennes x_i qui va satisfaire toutes les clauses C_i de $F_{n,m}$.

1. On note par $|C_i|$ la longueur de la clause C_i. Montrer que si $\sum_i^m 2^{-|C_i|} < 1$ alors $F_{n,m}$ est SAT.

∗. On rappele que $g(n) = \Omega(f(n))$ se lit "g grandit au moins aussi vite que f" ou encore "l'ordre de grandeur de g est au moins celui de f".