Université Joseph Fourier
GRENOBLE
UFR IM²AG

L3-INFO Algorithmique et Modélisation

Quick – 4 mars 2015 – durée 1 h

Sont interdits : les documents, les ordinateurs, les téléphones (incluant smartphone, tablettes,... tout ce qui contient un dispositif électronique).
Seuls les dictionnaires papier pour les personnes de langue étrangère sont autorisés.

Exercice : Random-SAT

On se donne une expression booléenne sous forme normale conjonctive $\Phi(x_1, \cdots, x_n)$ portant sur n variables notées $\{x_1, x_2, \cdots x_n\}$, ayant m clauses, chaque clause ayant k littéraux. C'est une instance du problème k-SAT.

$$\Phi(x_1, \cdots, x_n) = \bigwedge_{i=1}^{m} (x_{i_1} \vee x_{i_2} \cdots \vee x_{i_k});$$

avec x_{i_j} étant l'une des variables x_l ou sa négation. On suppose, sans perte de généralité qu'une clause ne contient pas deux fois la même variable.

Pour résoudre ce problème, on génère aléatoirement et uniformément les valeurs des n variables et on évalue les m clauses. On suppose que la génération des valeurs des variables x_i se fait de manière indépendantes et uniforme : probabilité $\frac{1}{2}$ d'avoir la valeur 1 (VRAI) et $\frac{1}{2}$ d'avoir la valeur 0 (FAUX).

Soit Z_i la variable aléatoire modélisant la valeur de la clause i et soit N la variable aléatoire modélisant le nombre de clauses vérifiées, on a :

$$N = Z_1 + Z_2 + \cdots + Z_m.$$

1. Calculer la probabilité que la clause i ne soit pas vérifiée ($\mathbb{P}(Z_i = 0)$), en déduire la probabilité que la clause i soit vérifiée ($\mathbb{P}(Z_i = 1)$).
2. Calculer $\overline{m}$ le nombre moyen de clauses vérifiées.
3. En déduire que si $\overline{m} > m - 1$ alors la formule Φ est satisfiable.
   Indication : remarquer que $\overline{m} \leqslant m$
4. Sous la condition ci-dessus $\overline{m} > m - 1$, proposer un algorithme randomisé calculant une solution à $\Phi(x_1, \cdots, x_n) = 1$.
5. *(plus difficile)* Soit p la probabilité de générer une solution. Montrer que

   $$p.m + (1 - p)(m - 1) \geqslant \overline{m}.$$

   En déduire un minorant de p. Donner un majorant de la moyenne du nombre de tirages nécessaires pour trouver une solution.
6. Que pensez-vous de cette approche ?

Exercice : SUM

Soit $\mathcal{E}$ un ensemble de n éléments. À chaque élément i de $\mathcal{E}$ on associe une valeur entière v_i. Pour éviter les cas particuliers on supposera les v_i distincts.

On recherche dans un premier temps tous les couples (i, j) avec $i \neq j$ tels que

$v_i + v_j = S$ pour un S donné ;

ou *pas de couple* s'il n'en existe pas.

1. Écrire un algorithme naïf qui résout le problème en $\mathcal{O}(n^2)$ opérations.
2. Écrire un algorithme qui résout le problème en $\mathcal{O}(n \log n)$ opérations.
3. En utilisant une structure de donnée adaptée, écrire un algorithme qui résout le problème en $\mathcal{O}(n)$ opérations.

On recherche maintenant tous les sous-ensembles de $\mathcal{E}$ tels que la somme des valeurs des éléments du sous-ensemble soit égale à S, sans contrainte sur la taille des sous-ensembles. Un peu d'étude bibliographique indique que ce problème est dans $\mathcal{NP}$.

Cas où toutes les valeurs $\{v_i\}$ sont positives

4. En vous inspirant de l'algorithme d'énumération des parties proposer un algorithme qui visite tous les sous-ensembles de somme S. On précisera avec soin les conditions sous lesquelles on effectuera les appels récursifs.
5. Écrire la preuve de votre algorithme.

Cas où les valeurs v_i peuvent être positives ou négatives

6. Adapter l'algorithme précédent en changeant les conditions d'appels récursifs.