Examen – Devoir à la maison 2020.

Veuillez m'envoyer votre copie via email.
email : vlad@irif.fr
Subject : [ALGO AVANCEE] DM 2020 (Nom/Prénom/Filière/Numéro étudiant)
avant le 28 Mai 2020 à minuit.

Il sera tenu compte de la clarté de vos explications, de vos justifications, de vos exemples illustrés.

Exercice 1 : PRAM Recherche & comptage d'éléments

On considère le problème algorithmique suivant :
Entrées : un tableau T de n entiers et un éléments a.
Sortie : le nombre d'éléments de T égaux à a.

1. Ecrire un algorithme PRAM pour ce problème en spécifiant le modèle (EREW ou CREW ou CRCW), le nombre de processeurs (exemple n^2 ou n processeurs) et en justifiant la complexité de votre algorithme.

2. Donner un algorithme en temps $O(\log \log n)$ en utilisant n processeurs dans un modèle PRAM (que vous devez spécifier).

Exercice 2 : Algorithme randomizé pour 3-SAT

1. Rappeler le problème 3-SAT.

2. Donner un algorithme force-brute de résolution d'une instance 3-SAT. Quelle est sa complexité ?

Dans ce qui suit, on s'intéresse à une meilleure complexité en utilisant la randomization. On donne ci-dessous une itération d'un tel algorithme (du à Paturi, Pudlàk et Zane).

```
Data : Φ : une formule 3-SAT portant sur n variables (x_1, ··· , x_n) et m clauses.
Result : Une affectation α des x_i qui va satisfaire toutes les clauses OU l'assertion "UNSAT".
Prendre uniformément une permutation aléatoire π de {1, ··· , n};
for i = 1, ··· , n do
    if il existe une clause unitaire correspondant à x_π(i) then
        Satisfaire x_π(i) pour satisfaire la clause;
    else
        Mettre x_π(i) à VRAI ou FAUX avec probabilité 1/2;
    end
    Soit b l'affectation de x_π(i) faite précédemment;
    Φ := Φ[x_π(i) ← b], α_i := b;
end
if α satisfait toutes les clauses then
    return α                    (⋆ "SAT" avec les affectations des x_i dans α = (α_1, ··· , α_n) ⋆)
else
    return "UNSAT"              (⋆ on doit recommencer ⋆)
end
```

3. Une clause c est dite critique pour une variable x si c fixe la valeur de x. Montrer que s'il n'existe qu'une affectation UNIQUE des x_i pouvant satisfaire une formule Φ alors chaque variable impliquée dans la formule Φ a au moins une clause critique.

Dans tout ce qui suit, on supposera qu'**il existe une affectation unique des x_i satisfaisant la formule en entrée**.

4. Soit x une variable dont une clause critique est c. Montrer qu'avec probabilité $\frac{1}{3}$, x est la dernière variable impliquée dans c.

5. Soit G_π le nombre de variables correctement fixées par la permutation π de l'algorithme ci-dessus. Pourquoi a-t-on

$$\mathbb{E}[G_\pi] = \frac{2}{3}n\,?$$

6. Montrer (ou admettre que) que la probabilité de succès d'une itération de l'algorithme est au moins $\mathbb{E}[2^{-G_\pi}]$.

7. L'inégalité de Jensen stipule que pour une fonction convexe f sur un intervalle réel I et une variable aléatoire X à valeur dans I telle que $\mathbb{E}(X)$ existe, nous avons

$$f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)] \,.$$

Déduire que la probabilité qu'une itération de l'algorithme réussisse est d'au moins

$$2^{-2/3n} \,.$$

8. Ecrire alors un algorithme randomizé pour résoudre le problème 3-SAT. Quelle est sa complexité ? Comparer cette complexité avec celle de la question "**2.**" de l'exercice courant.

9. Donner les indications qui permettent de généraliser cette approche aux problèmes k-SAT (pour $k \geq 3$).

Exercice 3 : Maximum Independent Set

Dans cet exercice, on considère les graphes aléatoires $\mathbb{G}(n, \frac{1}{4})$ et le problème dit *Maximum Independent Set* ou MIS. Dans un graphe quelconque $G = (V, E)$ (V = ensemble des sommets, E = ensemble des arêtes), un sous-ensemble S de V, $S \subseteq V$ est dit indépendant ssi ses sommets sont deux à deux disjoints. Le problème MIS consiste à renvoyer la taille de l'ensemble indépendant de taille maximum d'un graphe G donné en entrée, i.e. retourner la valeur

$$\max_{S \subseteq V, \text{S est un ensemble indépendant}} \mathrm{Card}(S) \,.$$

1. Rappeler la définition de $\mathbb{G}(n, 1/4)$. Illustrer sur un petit exemple.

2. Donner un algorithme glouton mais exact retournant un MIS.

3. Quelle est la complexité de votre algorithme ?

4. Décrire le plus précisemment possible l'espérance de la valeur retournée par votre algorithme. (Suggestion : on pourra donner sans la résoudre une équation de récurrence satisfaite par cette espérance.)

5. On suppose maintenant qu'avec une probabilité tendant vers 1 quand n est grand, il existe un MIS de taille $c \log n$. Ecrire un algorithme capable de trouver cette valeur. Quelle est sa complexité ?

6. Que représentent les valeurs

$$\binom{n}{(c-0.01)\log n}$$

et

$$\binom{n}{c\log n} \times \left(\frac{3}{4}\right)^{\binom{c\log n}{2}} \qquad ?$$

7. En comparant $\binom{n}{(c-0.01)\log n}$ et $\binom{n}{c\log n}$, justifier pourquoi on aura une borne inférieure $n^{\Omega(\log n)}$ via l'approche de cet exercice.