Université des Sciences et de la Technologie Houari Boumediène
Faculté d'Electronique et d'Informatique – Département d'Informatique
1er année Master « Sécurité des Systèmes Informatiques » SSI - 2013/2014.

17 Novembre 2013

Complexité et Algorithmique avancée
« Premier Contrôle » durée 1h

Exercice 1 : (3 pts)

Considérons l'algorithme A pour lequel le nombre d'opération élémentaire dans le pire cas s'exprime comme suit :

$$T(n) = \sum_{i=1}^{n-1} T(i) + 1$$

Résoudre T(n) et en déduire l'ordre de complexité associé à l'algorithme A.

Solution :

$$T(n) = \sum_{i=1}^{n-1} T(i) + 1 \ = T(1) + T(2) + \cdots T(n-2) + T(n-1) + 1$$
$$= \left(\sum_{i=1}^{n-2} T(i) + 1\right) + T(n-1) = 2 * T(n-1)$$

Ainsi, T(n) = 2*T(n-1)

De la même manière on pourra démontrer que T(n-1) = 2*T(n-2).

donc, T(n) = 2*T(n-1) = 2*2 T(n-2) = 2^3 * T(n-3) = … = 2^n * T(1).

D'ici nous pouvons conclure que l'ordre de complexité associé à l'algorithme A est O(2^n) puisque T(1) est une constante.

Exercice 2 : (2 pts)

Soit f(n)=n*log n le nombre d'opérations élémentaires du pire cas d'un algorithme A.

1. Montrez que f(n) = O(n²).
2. A-t-on f(n) = Θ(n²) ? Justifier votre réponse.

Solution :

1. Déjà vu en cours
2. On ne peut pas avoir f(n) = Θ(n²), puisque la fonction logarithmique ne possède pas de borne inférieur en fonction de n². En utilisant la notation de landau vous allez constater que c ne peut être constante et tend vers 1/n ce qui confirme l'inexistence de cette borne inférieure.

Exercice 3 : (5 pt)

Soit T un tableau de permutation d'ordre N (dimension N), on s'intéresse à définir un ordonnancement des éléments de T de la manière suivante :

- Les nombres pairs seront rangés dans des positions paires
- Tous les nombres impairs seront rangés dans des positions impaires
- L'ordre d'apparition des nombres pairs (Respect. Impaires) dans le tableau initial doit être respecté.

9	7	2	1	4	3	5	6	8

…….

9	2	7	4	1	6	3	8	5

1. Proposer un algorithme qui produit cet ordonnancement
2. Déterminer le pire cas et en déduire la complexité de l'algorithme proposé ? Justifier votre réponse.
3. Comment peut-on réduire la complexité de l'algorithme proposé en (1). En déduire la complexité obtenue

Solution :

1. Algorithme

Plusieurs possibilités :

- Soit traiter les éléments du tableau un par un, si l'ordonnancement est respecté alors avancer sinon effectuer les permutations nécessaires pour mettre la valeur courante à la bonne place. Au pire cas cet algorithme est quadratique.
- Soit faire appel à un tableau auxiliaire, qui servira à stocker les valeurs paires. En suite une fusion entre les deux tableaux permettra d'effectuer l'ordonnancement demandé. Au pire cas cet algorithme est linéaire en temps mais nécessite un espace mémoire supplémentaire.

3. Pour réduire la complexité de l'algorithme proposé, on peut utiliser deux indices, le premier pour les cases paires et le second pour les cases impaires. Tant que l'ordonnancement est respecté (i.e, valeur paires dans une position paire et une valeur impaire dans une position impaire) avancer, sinon effectuer une permutation entre les valeurs des cases pointées par les deux indices. Au pire cas, cet algorithme est linaire et ne nécessite pas d'espace mémoire supplémentaire.

Bonne Chance !