



Concours d'accès au Doctorat 3 ième Cycle Informatique 2017– 2018

Le 29/10/2017

Matière 1 : Algorithmique avancée et complexité, Systèmes d'exploitation,
Coefficient 1, durée 1h30
(Spécialité : RSI)

Partie 1 : Algorithmique avancée et complexité

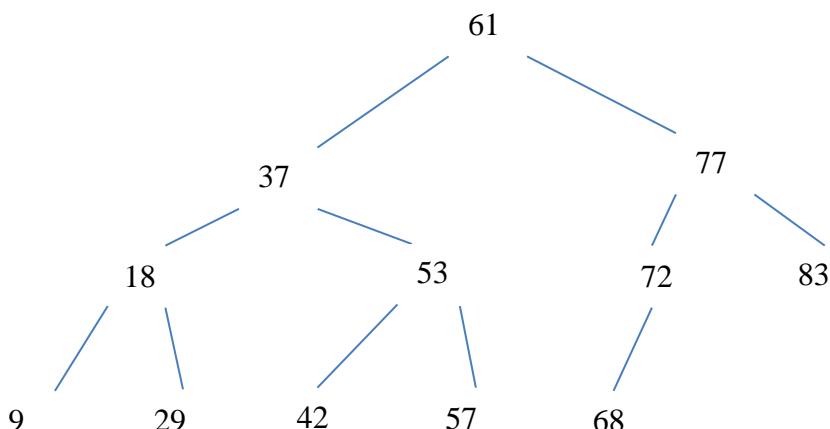
Exercice 1 : (14 points)

Soit $T[1..n]$ un tableau de n nombres réels non négatifs. Ce problème consiste à déterminer les indices $1 \leq i \leq j \leq n$ tels que le produit $T[i] * T[i + 1] * \dots * T[j]$ soit maximal. La solution à ce problème est un triplé (i, j, p_{\max}) (p_{\max} : le plus grand produit)

- Donner un algorithme naïf. Donner sa complexité.
- Donner un algorithme *Diviser pour régner* pour ce problème¹.
 - Donner l'équation de récurrence qui décrit le temps d'exécution de l'algorithme. Résoudre l'équation.
 - Est-il plus efficace que l'algorithme naïf ?

Exercice 2 : (6 points)

Considérer l'arbre binaire de recherche suivant :



- insérer le nombre 43 dans l'arbre.
- Ecrire un algorithme d'insertion d'un nombre dans un arbre de recherche. Calculer sa complexité.
- Ensuite supprimer le nombre 19 de l'arbre.
- Ecrire un algorithme de suppression d'un élément d'un arbre binaire de recherche. Calculer sa complexité.

Proposer un algorithme de tri en utilisant un arbre de recherche. Calculer sa complexité.

¹ On découpe le tableau en deux parties : le plus grand produit se trouve soit dans la partie de droite, soit dans la partie de gauche soit à cheval (chevauche) sur les deux parties. Dans ce cas il est constitué d'un plus grand produit de la partie gauche se terminant à la fin de la partie gauche, et d'un plus grand produit de la partie droite commençant au début de la partie droite.