

GÉNÉRALISATION D'UN RÉSULTAT DE COMPLEXITÉ À UNE CLASSE PLUS LARGE D'ALGORITHMES RÉCURSIFS

AMAR AHMANE & PIERRE-GABRIEL BERLUREAU

ABSTRACT. Si on note \mathcal{A} un algorithme prenant en entrée un seul paramètre de taille n . Dans l'hypothèse où \mathcal{A} est algorithme récursif utilisant la stratégie *diviser pour régner*, on montrera dans le Théorème I que, dans des conditions particulières, la complexité de l'algorithme est quasi-linéaire et on précisera la constante.

1. INTRODUCTION

Ce qui a motivé la recherche et la preuve du petit résultat généralisé qui sera exposé plus tard est l'exercice de Khôlle qui a été posé à Louis Thevenet. On considère l'algorithme suivant :

Algorithm 1

```
1: procedure FOO( $lst$ )
2:   if LENGTH( $lst$ ) = 0 then
3:     return 1
4:    $m \leftarrow \text{LENGTH}(lst)/3$ 
5:   return FOO( $lst[:m]$ ) + FOO( $lst[m:]$ )
```

On arrive très bien à dénombrer les différentes tailles qui apparaissent à un certain étage de l'arbre d'appel de cet algorithme récursif. On a alors le résultat suivant

Proposition 1. *L'Algorithme 1 a une complexité en $\mathcal{O}\left(n \log_{\frac{3}{2}}(n)\right)$.*

L'idée était alors de considérer le cas général décrit ci-dessous.

2. RÉSULTAT GÉNÉRALISÉ

Dans toute la suite, on notera \mathcal{A} un algorithme récursif utilisant la stratégie *diviser pour régner*. Un premier résultat que l'on va énoncer et démontrer ici consistera à dénombrer des objets d'une certaine forme dans un type d'arbres particulier. En réalité, on s'intéressera à des arbres à (au plus) m fils par noeud, avec $m \geq 2$ un entier naturel, définis par récurrence d'une façon qui sera précisée plus tard. Mais avant de s'y attaquer, précisons quelques notations et définitions locales.

Définition 2. Soit E un ensemble. On appelle arbre sur E un arbre dont les noeuds sont des éléments de E . On notera \mathfrak{A}_E l'ensemble des arbres sur E .

Définition 3. Soit E, F deux ensembles et $f \in F^E$. Alors, lorsque $A \in \mathfrak{A}_E$, on note $\tilde{f} : \mathfrak{A}_E \rightarrow \mathfrak{A}_F$ l'application qui à A associe un arbre sur F dont les noeuds sont les noeuds de A auxquels on aura appliqué f .

Proposition-Définition 4. Soit $m \geq 2$, $(x_1, \dots, x_m) \in (\mathbb{R}^*)^m$. On définit par récurrence un arbre parfait infini A sur \mathbb{R}^m de la façon suivante :

- La racine de A est $0_{\mathbb{R}^m}$.
- Pour tout noeud $N = (n_1, \dots, n_m)$ de A , N possède exactement m fils et pour tout $i \in \llbracket 1, m \rrbracket$, le i ème fils est le noeud $N_i = (n_1, \dots, n_i + x_i, \dots, n_m)$.

Ainsi, pour tout $k \in \mathbb{N}$, pour tout $(i_1, \dots, i_m) \in \mathbb{N}^m$ tel que $i_1 + \dots + i_m = k$, le nombre de noeuds de valeur $(i_1 x_1, i_2 x_2, \dots, i_m x_m)$ est

$$\frac{k!}{i_1! \dots i_m!}$$

Preuve test □

Théorème I. Soit $m \in \mathbb{N}^*$, $m \geq 2$, $p \in \mathbb{N}^* \setminus \{m\}$ et $(q_1, \dots, q_m) \in (\mathbb{N}^*)^m$ un m -uplet d'entiers non tous égaux tels que

$$\sum_{k=1}^m q_k = p$$

Soit \mathcal{A} un algorithme récursif prenant un seul paramètre et utilisant la stratégie diviser pour régner et tel que pour une entrée de taille n

$$T(n) = \sum_{k=1}^m T\left(\frac{q_k n}{p}\right) + \mathcal{O}(1)$$

où $T(n)$ représente le nombre d'opérations élémentaires effectuées par \mathcal{A} pour une entrée de taille n . Alors

$$T(n) = \mathcal{O}\left(n \log_{\frac{p}{\max(q_1, \dots, q_m)}}(n)\right)$$