



Exporter les résultats d'un algorithme Algobox dans un tableur.

publié le 11/01/2010 - mis à jour le 26/03/2010

Comparaison statistique des complexités de deux algorithmes de simplification des racines carrées.

Descriptif :

Comparaison statistique de deux algorithmes de simplification des racines carrées en Algobox.

Sommaire :

- Le problème :
- Extraction des résultats dans un tableur :
- Quelques conclusions :

● Le problème :

On souhaite créer un algorithme qui simplifie les racines carrées de nombres entiers sous la forme du produit d'un nombre entier par la racine d'un nombre entier le plus petit possible.

Après en avoir créé deux différents, on pourra comparer leur efficacité par une étude statistique du nombre de divisions effectuées pour les entiers de 1 à 1 000.

- Le premier algorithme consiste à trouver le plus grand carré qui divise le nombre :
 [Racines1](#) (ALG de 2.5 ko)
- Le deuxième algorithme teste lui aussi les carrés qui divisent le nombre mais il simplifie au fur et à mesure :
 [Racines2](#) (ALG de 2.9 ko)

● Extraction des résultats dans un tableur :

On modifie les deux algorithmes afin qu'ils effectuent les simplifications pour tous les nombres de 1 à 1 000 et qu'ils annoncent le nombre de divisions euclidiennes effectuées.

-  [Racines1Stats](#) (ALG de 2.8 ko)
-  [Racines2Stats](#) (ALG de 3.2 ko)

Pour utiliser les résultats dans un tableur, il suffit de sélectionner la liste des nombres, de presser **Ctrl C** (pour copier) puis **Ctrl V** (pour coller) dans une case du tableur, le tour est joué ! Vous pouvez maintenant utiliser les fonctions du tableur pour comparer les deux séries statistiques.

- Exemple d'étude :
 [RacinesStats](#) (OpenDocument Spreadsheet de 31.5 ko)

● Quelques conclusions :

On définit la **complexité** de ces algorithmes comme étant le nombre de divisions effectuées pour terminer l'algorithme. A chaque nombre dont on veut simplifier la racine carrée correspond donc une complexité. Prendre le nombre de divisions comme complexité est un choix, on pourrait prendre le temps d'exécution (qui dépend malheureusement de la machine utilisée), le nombre de conditions testées, le nombre d'instructions total (de tout

type) ou même encore la quantité de mémoire utilisée.

La complexité du premier algorithme est donnée par la partie entière de la racine carrée du nombre dont on simplifie la racine. Elle croît avec le nombre et lorsqu'on étudie pour les nombres de 1 à n , le pire cas est toujours celui pour n (et pour quelques prédécesseurs).

La complexité du deuxième est moins simple, elle est égale à celle du premier lorsque le nombre n'a pas de facteur carré mais est bien meilleure pour une puissance de 2 par exemple. La complexité n'est pas croissante, elle est inférieure ou égale à celle du premier. Le pire cas pour des nombres de 1 à n est presque le même mais il n'est pas toujours pour n .

Pour les nombres de 1 à 1 000, dans le cas médian, le deuxième algorithme a une complexité inférieure d'un tiers. Le gain en moyenne est un peu moins bon mais on gagne tout de même près de 25%.



**Académie
de Poitiers**

Avertissement : ce document est la reprise au format pdf d'un article proposé sur l'espace pédagogique de l'académie de Poitiers.

Il ne peut en aucun cas être proposé au téléchargement ou à la consultation depuis un autre site.