1 Analyse de l'algorithme de Karatsuba

On s'intéresse à la méthode de Karatsuba. On note $f(n)$ le nombre d'opérations arithmétiques sur les `double` effectuées.

Étape 1. Les deux questions qui suivent sont destinées à établir la fonction $f_{\text{simplifie}}(n)$, qui donne le nombre d'opérations sur les `double` par l'algorithme de Karatsuba, dans le cas simplifié $n = 2^k$.

Question 1. En analysant le code de la fonction `Karatsuba` de la feuille de TD, écrire une relation de récurrence avec condition initiale, pour la méthode de Karatsuba, dans le cas simplifié où $n = 2^k$. Faire attention à la condition initiale (multiplication élémentaire de deux polynômes constants). L'intuition peut être trompeuse. Bien regarder le code donné sur la feuille de TD.

Question 2. Résoudre cette relation de récurrence avec condition initiale, en utilisant le logiciel MAPLE (disponible sur le serveur `weppes`). Dans la suite de cet énoncé, on appellera $f_{\text{simplifie}}(n)$ la formule ainsi obtenue. Compléter :

$$f_{\text{simplifie}}(n) =$$

Pour vous aider, voici la résolution, avec MAPLE, d'une autre relation de récurrence avec condition initiale.

```
fboulier@weppes$ /usr/local/maple11/bin/maple
> syst := { f(n) = 2*f(n-1) + 3, f(0) = 1 };
                   syst := {f(n) = 2 f(n - 1) + 3, f(0) = 1}

> rsolve (syst, f(n));
                                        n
                                     4 2  - 3
```

Étape 2. Les questions qui suivent sont destinées à fabriquer un fichier de mesures (une courbe mesurée) donnant le nombre d'opérations flottantes sur les `double` effectuées par une vraie implantation de l'algorithme de Karatsuba, pour des valeurs quelconques de n.

Question 3. Télécharger le fichier `"Karatsuba.c"`. Le compiler et l'exécuter. Pour le compiler, il est nécessaire d'éditer les liens avec la bibliothèque mathématique (option `-lm`).

```
$ wget --no-cache http://www.lifl.fr/~boulier/polycopies/SD/Karatsuba.c
$ gcc Karatsuba.c -lm
```

Question 4. Modifier le programme pour qu'il compte, à l'exécution, le nombre d'opérations arithmétiques sur les `double` effectuées par l'algorithme de Karatsuba. Afficher les mesures sur deux colonnes : une pour n et une pour $f(n)$. Aller jusqu'à $n = 600$. Enregistrer le résultat dans un fichier `"Karatsuba.stats"`. Voici un exemple (les valeurs numériques sont fausses) :

```
# n f(n)
  0 1
  1 5
  2 13
  3 29
  4 61
  5 125
  6 253
  7 509
  8 1021
  9 2045
```

Question 5. Même question, mais pour la multiplication élémentaire. Le fichier doit s'appeler `"Elementaire.stats"`.

Question 6. Chacun de ces deux fichiers définit une courbe (une courbe « mesurée »). Tracer sur un même graphique, avec GNUPLOT, les deux courbes mesurées. Quel est le meilleur algorithme ? À partir de quelle valeur de n est-il meilleur ?

Pour vous aider, voici quelques commandes GNUPLOT.

```
$ gnuplot
gnuplot> plot "fichier1" with lines, "fichier2" with lines
gnuplot> plot [0:100] "fichier1" with lines, 2*x**2
```

Étape 3. Les questions qui suivent sont destinées à se convaincre, au moins expérimentalement, que le nombre $f(n)$ d'opérations sur les `double` effectuées, dans tous les cas, par l'algorithme de Karatsuba, est bien un équivalent asymptotique de la fonction $f_{\text{simplifie}}(n)$ établie dans le cas simplifié uniquement.

Question 7. Tracer sur un même graphique, avec GNUPLOT, la courbe mesurée correspondant à l'algorithme de Karatsuba et la courbe théorique du cas simplifié $f_{\text{simplifie}}(n)$ (attention, avec GNUPLOT, la variable doit être appelée x et pas n). Normalement, les deux courbes doivent se croiser pour tout n de la forme 2^k. Vérifier.

Question 8. Où se situe la courbe théorique du cas simplifié $f_{\text{simplifie}}(n)$ par rapport à la courbe mesurée $f(n)$? En faisant confiance au graphique, peut-on affirmer que $f(n) \in \Omega(n^{\log_2(3)})$? que $f(n) \in O(n^{\log_2(3)})$? que $f(n) \in \Theta(n^{\log_2(3)})$?

Question 9. Pour finir de se convaincre que $f(n) \in \Theta(n^{\log_2(3)})$, on peut appliquer deux méthodes :

1. Chercher empiriquement une autre courbe $F(n)$, asymptotiquement équivalent à $f_{\text{t}}(n)$, située de « l'autre côté » de la courbe mesurée.
2. Tester la courbe $F(n) \underset{\text{def}}{=} f(2\,n)$. Développer la formule avec MAPLE.

Étape 4. Cette étape apporte quelques compléments aux questions précédentes.

Question 10. S'il fallait approximer la courbe mesurée par une formule de la forme $a\, n^{\log_2(3)}$, quelle serait la meilleure valeur pour a ? Une solution consiste à utiliser la fonction `fit` de GNU-PLOT. Quelle valeur obtient-on ?

$$a =$$

```
$ gnuplot
gnuplot> log2(x)=log(x)/log(2)
gnuplot> fit a*x**log2(3)+b*x+c "fichier" via a,b,c
```

Question 11. Dans notre implantation de l'algorithme de Karatsuba, on appelle la multiplication élémentaire quand $n = 1$. Or, la multiplication élémentaire reste meilleure que l'algorithme de Karatsuba pour des valeurs de n plus élevées que 1.

Modifier en conséquence le seuil à partir duquel l'algorithme de Karatsuba devrait appeler la multiplication élémentaire.

Le seuil optimal n'est pas celui qu'on a déterminé graphiquement quelques questions plus haut. Voyez-vous pourquoi ? Déterminer ce seuil expérimentalement.

Le nouvel algorithme est-il toujours équivalent asymptotiquement à $n^{\log_2(3)}$? Si oui, quelle nouvelle valeur donner à la constante a de la question précédente ?