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.

Solution. Ça a été fait en TD. On trouve

$$f(n) = 3\,f\left(\frac{n}{2}\right) - 4\,n + 3\,, \quad f(1) = 2\,.$$

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
```

Solution. Voir ci-dessous. On trouve donc

$$f(n) = \frac{17}{2}\,n^{\log_2(3)} - 8\,n + \frac{3}{2}.$$

```
 syst := { f(n) = 3*f(n/2) + 4*n - 3, f(1)=2};
                syst := {f(1) = 2, f(n) = 3 f(n/2) + 4 n - 3}

> rsolve (syst, f(n));
                                /ln(3)\
                                |-----|
                                \ln(2)/
                          17/2 n        + 3/2 - 8 n

> evalf (%);
                             1.584962501
                8.500000000 n            + 1.500000000 - 8. n
```

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
```

Solution. Ça a été préparé en TD. Pour chaque fonction `add_poly`, `sub_poly`, `mul_poly` et `Karatsuba`, il faut changer le type de retour `void` en `int`, rajouter un compteur local c initialisé à zéro et compter les opérations sur les doubles. Il faut modifier aussi le programme principal. Pour avoir un bon fichier de mesures, prendre $N = 2050$. Voir fichier en attachement. Obtenir le résultat en affichant sur la sortie standard et en redirigeant la sortie. Le fichier de données commence comme ça.

```
1 2
2 11
3 33
4 46
5 94
6 120
...
1023 493681
1024 493726
```

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

Solution. Le fichier de données commence comme ça.

```
1 2
2 8
3 18
4 32
5 50
6 72
```

```
...
1023 2093058
1024 2097152
```

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 ?

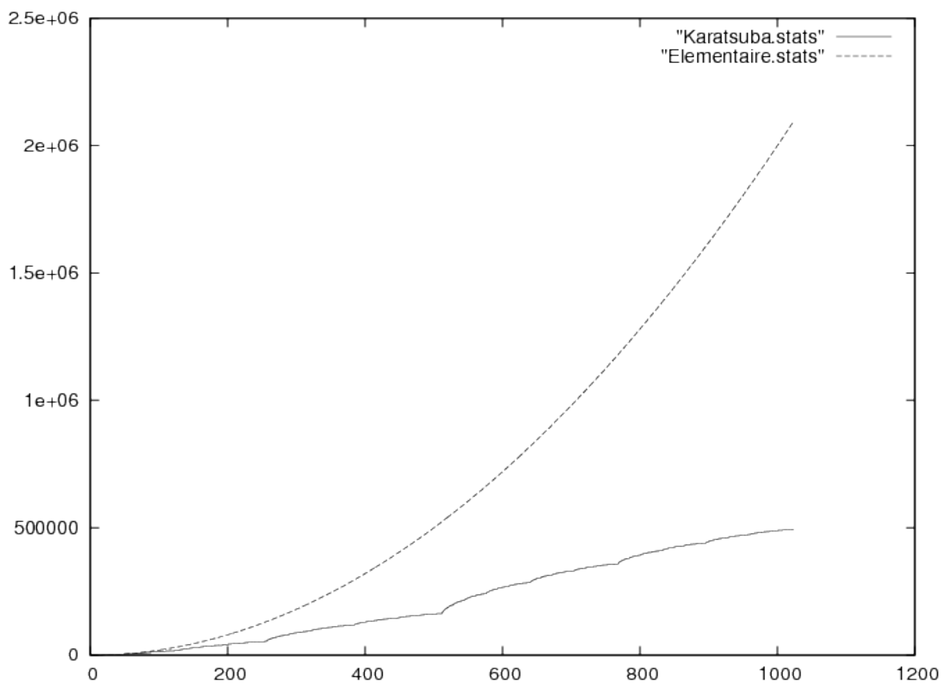
Pour vous aider, voici quelques commandes GNUPLOT.

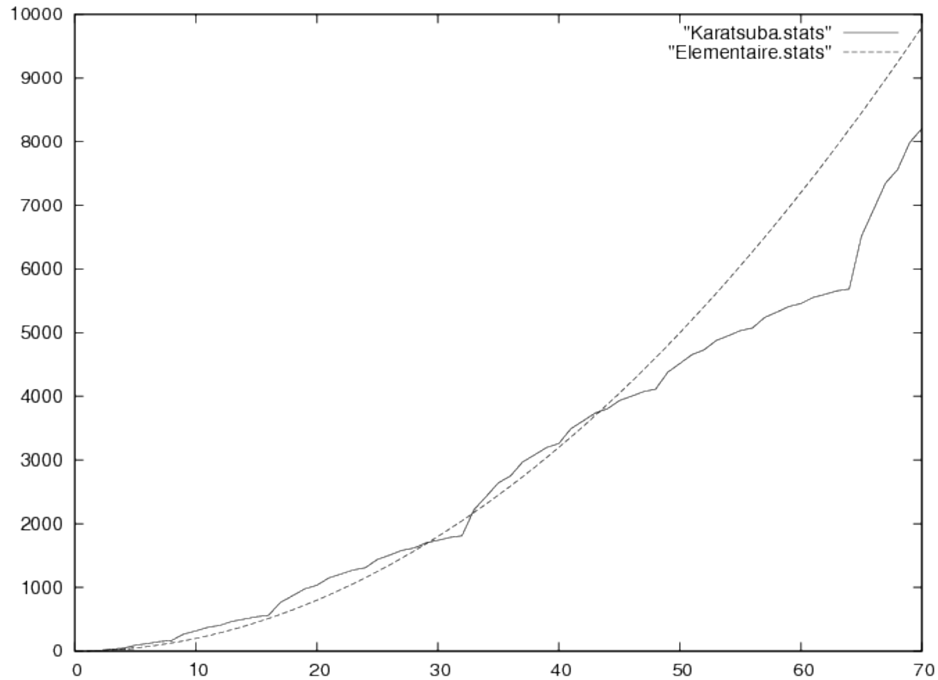
```
$ gnuplot
gnuplot> plot "fichier1" with lines, "fichier2" with lines
gnuplot> plot [0:100] "fichier1" with lines, 2*x**2
```

SOLUTION. La méthode de Karatsuba commence à devenir meilleure vers $n \simeq 43$. Voici les commandes et les graphiques.

```
plot "Karatsuba.stats" with lines, "Elementaire.stats" with lines
plot [0:70] "Karatsuba.stats" with lines, "Elementaire.stats" with lines

Pour la génération des graphiques:
set terminal postscript
set output "courbe1.ps"
plot "Karatsuba.stats" with lines, "Elementaire.stats" with lines
set output "courbe2.ps"
plot [0:70] "Karatsuba.stats" with lines, "Elementaire.stats" with lines
```



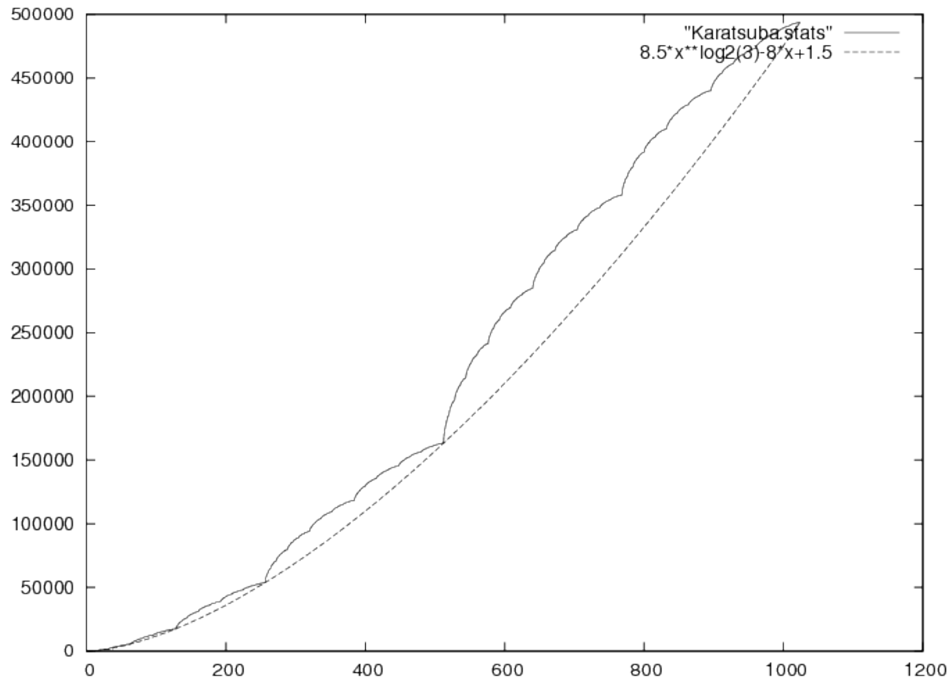


Étape 3. Les questions qui suivent sont destinées à se convaincre, au moins expérimentalement, que le nombre $f(n)$ d'opérations sur les `double` effectuées, dans tous les cas, par l'algorithme de Karatsuba, est bien un équivalent asymptotique de la fonction $f_{\text{simplifie}}(n)$ établie dans le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

Solution. Voici les commandes et les graphiques.

```
log2(x)=log(x)/log(2)
plot "Karatsuba.stats" with lines, 8.5*x**log2(3)-8*x+1.5
```



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 $f(n) \in \Theta(n^{\log_2(3)})$?

> Solution. La courbe est en dessous. Si on fait confiance au graphique, on peut affirmer que $f(n) \in \Omega(n^{\log_2(3)})$.

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.

> Solution. Si on remplace le coefficient 17/2 par 11, il semble que la courbe soit au-dessus.

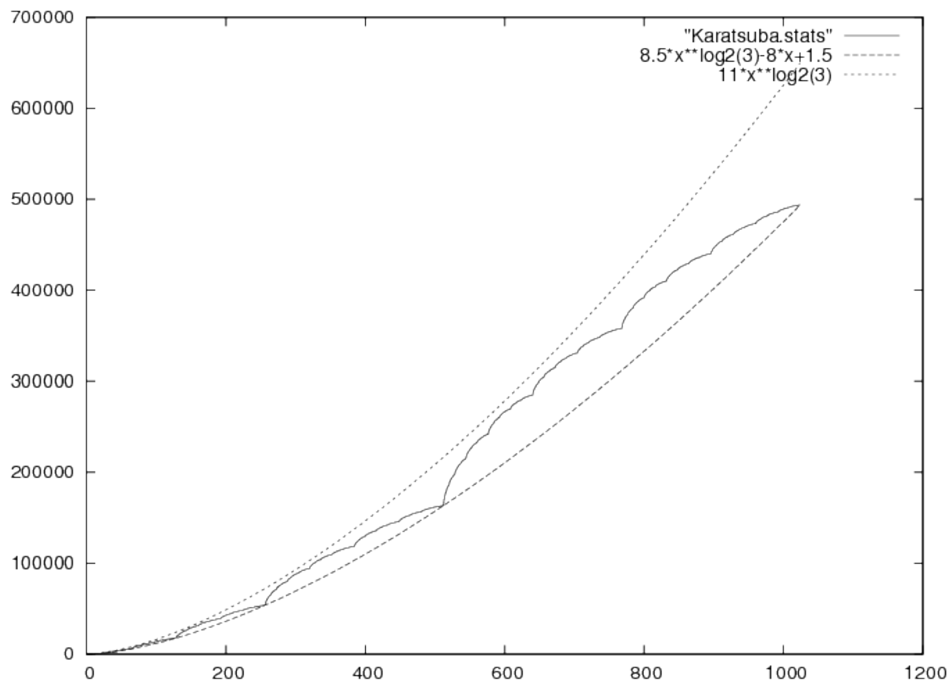
```
log2(x)=log(x)/log(2)
plot "Karatsuba.stats" with lines, 8.5*x**log2(3)-8*x+1.5, 11*x**log2(3)
```

Avec MAPLE, on trouve bien une courbe de la forme $a\,n^{\log_2(3)}$.

```
> f := 17/2*n^(log[2](3)) - 8*n + 3/2;
                                   /ln(3)\
                                   |-----|
                                   \ln(2)/
                        f := 17/2 n        - 8 n + 3/2

> F := subs (n = 2*n, f);
                                     /ln(3)\
                                     |-----|
                                     \ln(2)/
                      F := 17/2 (2 n)        - 16 n + 3/2
```

```
> simplify (%);
                                /ln(3)\
                                |-----|
                                \ln(2)/
                          51/2 n        - 16 n + 3/2
```



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 GNUPLOT. 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
```

Solution. Voici la solution avec gnuplot. On trouve $a \simeq 6.13$

```
log2(x)=log(x)/log(2)
fit a*x**log2(3)+b*x+c "Karatsuba.stats" via a,b,c

Final set of parameters            Asymptotic Standard Error
=======================            ==========================

a               = 6.12836          +/- 0.1542       (2.516%)
b               = 186.537          +/- 9.252        (4.96%)
c               = -17284.9         +/- 1457         (8.431%)
```

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 ?

Solution. Pas fait.