Algorithmes efficaces pour les grands nombres et polynômes : Partie 1

Salem BENFERHAT

Centre de Recherche en Informatique de Lens (CRIL-CNRS)
email : benferhat@cril.fr

Motivations

- Peut-on calculer :

Motivations

- Peut-on calculer :
  - le produit éde deux matrices carrées $n \times n$ en moins de $O(n^3)$?

Motivations

- Peut-on calculer :
  - le produit éde deux matrices carrées $n \times n$ en moins de $O(n^3)$?
  - le produit de deux grands nombres, chacun de taille n, en moins de $O(n^2)$?

Motivations

- Peut-on calculer :
  - le produit éde deux matrices carrées $n \times n$ en moins de $O(n^3)$?
  - le produit de deux grands nombres, chacun de taille n, en moins de $O(n^2)$?
  - le produit de deux polynômes, chacun de degré n, en moins de $O(n^2)$?

Motivations

- Peut-on calculer :
  - le produit éde deux matrices carrées $n \times n$ en moins de $O(n^3)$?
  - le produit de deux grands nombres, chacun de taille n, en moins de $O(n^2)$?
  - le produit de deux polynômes, chacun de degré n, en moins de $O(n^2)$?

Impact

Toute réduction de complexité a un impact sur de nombreuses applications comme le calcul scientifique ou encore la cryptographie.

Commençons par

les nombres complexes

Les nombres complexes

Définition

Un nombre complexe, de la forme $x + i.y$, contient deux parties :

- Une partie réelle x
- Une partie imaginaire y

Les nombres complexes

Définition

Un nombre complexe, de la forme $x + i.y$, contient deux parties :

- Une partie réelle x
- Une partie imaginaire y

Le terme i est au fait une valeur imaginaire et représente une solution à l'équation $i^2 = -1$

Nombre complexe : Déclaration

```
2 typedef struct
3     {
4      float reelle;
5      float  complexe;
6     } nbrecomplexe;
```

Nombres complexes

Exercice

Soit z et w deux nombres complexes. Ecrire une fonction qui :

- calcule la somme de z et w,

Nombres complexes

Exercice

Soit z et w deux nombres complexes. Ecrire une fonction qui :

- calcule la somme de z et w,
- calcule le produit de z et w, et

Nombres complexes

Exercice

Soit z et w deux nombres complexes. Ecrire une fonction qui :

- calcule la somme de z et w,
- calcule le produit de z et w, et
- calcule la soustraction de z et w.

Fonction : addition de deux nombres complexes

```
2
3  nbrecomplexe addition (nbrecomplexe z, nbrecomplexe w)
4  {
5    nbrecomplexe somme;
6    somme.reelle = z.reelle + w.reelle;
7    somme.complexe=z.complexe + w.complexe;
8    return somme;
9  }
```

Fonction : soustraction de deux nombres complexes

```
2
3  nbrecomplexe soustraction (nbrecomplexe z, nbrecomplexe w)
4  {
5    nbrecomplexe moins;
6    moins.reelle = z.reelle - w.reelle;
7    moins.complexe=z.complexe - w.complexe;
8    return moins;
9  }
```

Fonction : produit de deux nombres complexes

```
2  nbrecomplexe produit (nbrecomplexe z, nbrecomplexe w)
3  {
4    nbrecomplexe mult;
5    mult.reelle = (z.reelle * w.reelle) - (z.complexe * w.complexe);
6    mult.complexe=z.reelle * w.complexe + w.reelle * z.complexe;
7    return mult;
8  }
```

Complexité du produit de deux nombres complexes

Remarques

- Le produit de deux grands nombres complexes nécessite :

Complexité du produit de deux nombres complexes

Remarques

- Le produit de deux grands nombres complexes nécessite :
  - 4 opérations élémentaires de type ”multiplication”

Complexité du produit de deux nombres complexes

Remarques

- Le produit de deux grands nombres complexes nécessite :
  - 4 opérations élémentaires de type ”multiplication”
  - Une opération élémentaire de type ”soustraction”

Complexité du produit de deux nombres complexes

Remarques

- Le produit de deux grands nombres complexes nécessite :
  - 4 opérations élémentaires de type ”multiplication”
  - Une opération élémentaire de type ”soustraction”
  - Une opération élémentaire de type ”addition”

Complexité du produit de deux nombres complexes

Remarques

- Si ces quatre opérations s'appliquent sur des types simples (comme ici, de type standard "réel") alors il est clair que la complexité temporelle du produit de deux nombres complexes sera en $O(1)$.

Complexité du produit de deux nombres complexes

Remarques

- Si ces quatre opérations s'appliquent sur des types simples (comme ici, de type standard "réel") alors il est clair que la complexité temporelle du produit de deux nombres complexes sera en $O(1)$.
- Cette complexité ne sera plus en $O(1)$ si les opérations arithmétiques (*, +, -) s'appliquent sur des tableaux par exemples. En effet, dans ce cas, les opérations de multiplications sont plus coûteuses que celles des additions (ou des soustractions).

Complexité du produit de deux nombres complexes

Peut-on faire mieux?

- Dans l'hypothèse où le produit est plus coûteux que l'addition (ou la soustraction), peut-on améliorer le calcul de produits de nombres complexes?

Complexité du produit de deux nombres complexes

Peut-on faire mieux?

- Dans l'hypothèse où le produit est plus coûteux que l'addition (ou la soustraction), peut-on améliorer le calcul de produits de nombres complexes?
- L'idée est d'essayer d'obtenir le même résultat avec moins d'opérations de multiplication (quitte à avoir plus d'opérations d'addition et de soustraction).

Une simple reformulation ...

Soit $z = z_r + i * z_c$ et $w = w_r + i * w_c$ deux nombres complexes.

Une simple reformulation ...

Soit $z = z_r + i * z_c$ et $w = w_r + i * w_c$ deux nombres complexes.
Soit :

$$p_1 = z_r * w_r$$

Une simple reformulation ...

Soit $z = z_r + i * z_c$ et $w = w_r + i * w_c$ deux nombres complexes.
Soit :

$$\begin{aligned} p_1 &= z_r * w_r \\ p_2 &= z_c * w_c \end{aligned}$$

Une simple reformulation ...

Soit $z = z_r + i * z_c$ et $w = w_r + i * w_c$ deux nombres complexes.
Soit :

$$\begin{aligned} p_1 &= z_r * w_r \\ p_2 &= z_c * w_c \\ p_3 &= (z_r + z_c) * (w_r + w_c) \end{aligned}$$

Une simple reformulation ...

Soit $z = z_r + i * z_c$ et $w = w_r + i * w_c$ deux nombres complexes.
Soit :

$$\begin{aligned} p_1 &= z_r * w_r \\ p_2 &= z_c * w_c \\ p_3 &= (z_r + z_c) * (w_r + w_c) \end{aligned}$$

Question

Reformuler :

$$z * w = (z_r * w_r - z_c * w_c) + i * (z_r * w_c + z_c * w_r)$$

en fonction de p_1, p_2, p_3.

Une simple reformulation ...

Soit $z * w = (z_r * w_r - z_c * w_c) + i * (z_r * w_c + z_c * w_r)$ et :

$$\begin{aligned} p_1 &= z_r * w_r \\ p_2 &= z_c * w_c \\ p_3 &= (z_r + z_c) * (w_r + w_c) \end{aligned}$$

Une simple reformulation ...

Soit $z * w = (z_r * w_r - z_c * w_c) + i * (z_r * w_c + z_c * w_r)$ et :

$$\begin{aligned} p_1 &= z_r * w_r \\ p_2 &= z_c * w_c \\ p_3 &= (z_r + z_c) * (w_r + w_c) \end{aligned}$$

Réponse

$$z * w = (p_1 - p_2) + i * (p_3 - p_2 - p_1)$$

Avec cette formulation ...

$$\begin{aligned} p_1 &= z_r * w_r \\ p_2 &= z_c * w_c \\ p_3 &= (z_r + z_c) * (w_r + w_c) \\ z * w &= (p_1 - p_2) + i.(p_3 - p_2 - p_1) \end{aligned}$$

Nous effectuons :

Avec cette formulation ...

$$\begin{aligned} p_1 &= z_r * w_r \\ p_2 &= z_c * w_c \\ p_3 &= (z_r + z_c) * (w_r + w_c) \\ z * w &= (p_1 - p_2) + i.(p_3 - p_2 - p_1) \end{aligned}$$

Nous effectuons :

- 3 multiplications

Avec cette formulation ...

$$\begin{aligned} p_1 &= z_r * w_r \\ p_2 &= z_c * w_c \\ p_3 &= (z_r + z_c) * (w_r + w_c) \\ z * w &= (p_1 - p_2) + i.(p_3 - p_2 - p_1) \end{aligned}$$

Nous effectuons :

- 3 multiplications
- 3 additions, et

Avec cette formulation ...

$$\begin{aligned} p_1 &= z_r * w_r \\ p_2 &= z_c * w_c \\ p_3 &= (z_r + z_c) * (w_r + w_c) \\ z * w &= (p_1 - p_2) + i.(p_3 - p_2 - p_1) \end{aligned}$$

Nous effectuons :

- 3 multiplications
- 3 additions, et
- 3 soustractions.

Avec cette formulation ...

$$\begin{aligned} p_1 &= z_r * w_r \\ p_2 &= z_c * w_c \\ p_3 &= (z_r + z_c) * (w_r + w_c) \\ z * w &= (p_1 - p_2) + i.(p_3 - p_2 - p_1) \end{aligned}$$

Nous effectuons :

- 3 multiplications
- 3 additions, et
- 3 soustractions.

Donc un gain en terme de complexité calculatoire si l'opération de multiplication est plus coûteuse que les opérations d'addition et de soustraction.

Produit de deux grands nombres

Question

Peut-on exploiter cette idée pour réduire la complexité du calcul du produit de deux grands nombres?

Produit de deux grands nombres

Principe

- Soit A un grand nombre de taille N. Supposons $N = 2^m$ une puissance de 2 (il ne s'agit pas d'une restriction, il suffit si besoin de compléter par des zéros)

Produit de deux grands nombres

Principe

- Soit A un grand nombre de taille N. Supposons $N = 2^m$ une puissance de 2 (il ne s'agit pas d'une restriction, il suffit si besoin de compléter par des zéros)
- Notons :
  - A_s contient les $\frac{N}{2}$ premiers chiffres du poids les plus forts

Produit de deux grands nombres

Principe

- Soit A un grand nombre de taille N. Supposons $N = 2^m$ une puissance de 2 (il ne s'agit pas d'une restriction, il suffit si besoin de compléter par des zéros)
- Notons :
  - A_s contient les $\frac{N}{2}$ premiers chiffres du poids les plus forts
  - A_i contient les $\frac{N}{2}$ derniers chiffres du poids les plus faibles

Produit de deux grands nombres

Principe

- Soit A un grand nombre de taille N. Supposons $N = 2^m$ une puissance de 2 (il ne s'agit pas d'une restriction, il suffit si besoin de compléter par des zéros)
- Notons :
  - A_s contient les $\frac{N}{2}$ premiers chiffres du poids les plus forts
  - A_i contient les $\frac{N}{2}$ derniers chiffres du poids les plus faibles

Produit de deux grands nombres

Principe

- Soit A un grand nombre de taille N. Supposons $N = 2^m$ une puissance de 2 (il ne s'agit pas d'une restriction, il suffit si besoin de compléter par des zéros)
- Notons :
  - A_s contient les $\frac{N}{2}$ premiers chiffres du poids les plus forts
  - A_i contient les $\frac{N}{2}$ derniers chiffres du poids les plus faibles
- Un nombre A peut-être alors décomposé comme suit :

$$A = A_s * 10^{\frac{N}{2}} + A_i$$

Produit de deux grands nombres

Exemple

- Soit A = 18730092. $N = 8$

Produit de deux grands nombres

Exemple

- Soit A = 18730092. $N = 8$
- Notons :
  - $A_s = 1873$

Produit de deux grands nombres

Exemple

- Soit A = 18730092. $N = 8$
- Notons :
  - $A_s = 1873$
  - $A_i = 0092$

Produit de deux grands nombres

Exemple

- Soit A = 18730092. $N = 8$
- Notons :
  - $A_s = 1873$
  - $A_i = 0092$
- Le nombre A = 18730092 peut-être alors décomposé comme suit :

$$A = 1873 * 10^4 + 0092$$

Produit de deux grands nombres

Principe

- Soit $A = (A_s * 10^{\frac{N}{2}} + A_i)$ et $B = (B_s * 10^{\frac{N}{2}} + B_i)$ deux grands nombres de taille N.

Produit de deux grands nombres

Principe

- Soit $A = (A_s * 10^{\frac{N}{2}} + A_i)$ et $B = (B_s * 10^{\frac{N}{2}} + B_i)$ deux grands nombres de taille N.
- Le but est de calculer $A * B$

Produit de deux grands nombres

Principe

- Soit $A = (A_s * 10^{\frac{N}{2}} + A_i)$ et $B = (B_s * 10^{\frac{N}{2}} + B_i)$ deux grands nombres de taille N.
- Le but est de calculer $A * B$
- Ce problème est similaire à celui du produit de deux nombres complexes, à deux détails près :

Produit de deux grands nombres

Principe

- Soit $A = (A_s * 10^{\frac{N}{2}} + A_i)$ et $B = (B_s * 10^{\frac{N}{2}} + B_i)$ deux grands nombres de taille N.
- Le but est de calculer $A * B$
- Ce problème est similaire à celui du produit de deux nombres complexes, à deux détails près :
  - i est remplacé par $10^{\frac{N}{2}}$, et de ce fait on obtient des entiers

Produit de deux grands nombres

Principe

- Soit $A = (A_s * 10^{\frac{N}{2}} + A_i)$ et $B = (B_s * 10^{\frac{N}{2}} + B_i)$ deux grands nombres de taille N.
- Le but est de calculer $A * B$
- Ce problème est similaire à celui du produit de deux nombres complexes, à deux détails près :
  - i est remplacé par $10^{\frac{N}{2}}$, et de ce fait on obtient des entiers
  - Les 3 multiplications nécessaires pour le calcul de $A * B$ peuvent-être à leur tour réitérées.

Produit de deux grands nombres

Principe

- Soit $A = (A_s * 10^{\frac{N}{2}} + A_i)$ et $B = (B_s * 10^{\frac{N}{2}} + B_i)$ deux grands nombres de taille N.
- Le but est de calculer $A * B$
- Ce problème est similaire à celui du produit de deux nombres complexes, à deux détails près :
  - i est remplacé par $10^{\frac{N}{2}}$, et de ce fait on obtient des entiers
  - Les 3 multiplications nécessaires pour le calcul de $A * B$ peuvent-être à leur tour réitérées.

Produit de deux grands nombres

Décomposition ...

- Soit $A = (A_s * 10^{\frac{N}{2}} + A_i)$ et $B = (B_s * 10^{\frac{N}{2}} + B_i)$

Produit de deux grands nombres

Décomposition ...

- Soit $A = (A_s * 10^{\frac{N}{2}} + A_i)$ et $B = (B_s * 10^{\frac{N}{2}} + B_i)$
- Définissons :
  - $p_1 = A_s * B_s$
  - $p_2 = A_i * B_i$
  - $p_3 = (A_s + A_i) * (B_s + B_i)$

Produit de deux grands nombres

Décomposition ...

- Soit $A = (A_s * 10^{\frac{N}{2}} + A_i)$ et $B = (B_s * 10^{\frac{N}{2}} + B_i)$ et :
  - $p_1 = A_s * B_s$
  - $p_2 = A_i * B_i$
  - $p_3 = (A_s + A_i) * (B_s + B_i)$

Produit de deux grands nombres

Décomposition ...

- Soit $A = (A_s * 10^{\frac{N}{2}} + A_i)$ et $B = (B_s * 10^{\frac{N}{2}} + B_i)$ et :
  - $p_1 = A_s * B_s$
  - $p_2 = A_i * B_i$
  - $p_3 = (A_s + A_i) * (B_s + B_i)$

Remarque 1

Il est alors facile de vérifier que :

$$A * B = (p_1 * 10^N) + p_2 + (p_3 - p_1 - p_2) * 10^{\frac{N}{2}}$$

Produit de deux grands nombres

Décomposition ...

- $p_1 = A_s * B_s$, $p_2 = A_i * B_i$
- $p_3 = (A_s + A_i) * (B_s + B_i)$
- $A * B = (p_1 * 10^N) + p_2 + (p_3 - p_1 - p_2) * 10^{\frac{N}{2}}$.

Produit de deux grands nombres

Décomposition ...

- $p_1 = A_s * B_s$, $p_2 = A_i * B_i$
- $p_3 = (A_s + A_i) * (B_s + B_i)$
- $A * B = (p_1 * 10^N) + p_2 + (p_3 - p_1 - p_2) * 10^{\frac{N}{2}}$.

Remarque 2

L'expression : $Y * 10^m$

- consiste tout simplement à insérer m zéros à droite du grand nombre Y.

Produit de deux grands nombres

Décomposition ...

- $p_1 = A_s * B_s$, $p_2 = A_i * B_i$
- $p_3 = (A_s + A_i) * (B_s + B_i)$
- $A * B = (p_1 * 10^N) + p_2 + (p_3 - p_1 - p_2) * 10^{\frac{N}{2}}$.

Remarque 2

L'expression : $Y * 10^m$

- consiste tout simplement à insérer m zéros à droite du grand nombre Y.
- cette opération se fait en $O(Taille)$ où Taille est la taille des grands nombres manipulés.

Produit de deux grands nombres

Décomposition ...

- $p_1 = A_s * B_s$, $p_2 = A_i * B_i$
- $p_3 = (A_s + A_i) * (B_s + B_i)$
- $A * B = (p_1 * 10^N) + p_2 + (p_3 - p_1 - p_2) * 10^{\frac{N}{2}}$.

Produit de deux grands nombres

Décomposition ...

- $p_1 = A_s * B_s$, $p_2 = A_i * B_i$
- $p_3 = (A_s + A_i) * (B_s + B_i)$
- $A * B = (p_1 * 10^N) + p_2 + (p_3 - p_1 - p_2) * 10^{\frac{N}{2}}$.

Remarque 3

Les trois multiplications p_1, p_2, p_3 nécessaires pour le calcul de $A * B$ peuvent se faire à leur tour de manière récursive. C'est-à-dire que le calcul du produit de deux grands nombres de taille N revient à faire :

Produit de deux grands nombres

Décomposition ...

- $p_1 = A_s * B_s$, $p_2 = A_i * B_i$
- $p_3 = (A_s + A_i) * (B_s + B_i)$
- $A * B = (p_1 * 10^N) + p_2 + (p_3 - p_1 - p_2) * 10^{\frac{N}{2}}$.

Remarque 3

Les trois multiplications p_1, p_2, p_3 nécessaires pour le calcul de $A * B$ peuvent se faire à leur tour de manière récursive. C'est-à-dire que le calcul du produit de deux grands nombres de taille N revient à faire :

- Trois calculs du produit de deux grands nombres mais de taille $\frac{N}{2}$.

Produit de deux grands nombres

Décomposition ...

- $p_1 = A_s * B_s$, $p_2 = A_i * B_i$
- $p_3 = (A_s + A_i) * (B_s + B_i)$
- $A * B = (p_1 * 10^N) + p_2 + (p_3 - p_1 - p_2) * 10^{\frac{N}{2}}$.

Remarque 3

Les trois multiplications p_1, p_2, p_3 nécessaires pour le calcul de $A * B$ peuvent se faire à leur tour de manière récursive. C'est-à-dire que le calcul du produit de deux grands nombres de taille N revient à faire :

- Trois calculs du produit de deux grands nombres mais de taille $\frac{N}{2}$.
- Un certain nombre d'opérations (addition, soustraction, décalage) de complexité $O(N)$.

Produit de deux grands nombres

Notons :

- grand_nombre Produit (grand_nombre A, B)
  - Une fonction qui prend en entrée deux grands nombres A et B de taille N et retourne un grand nombre (de taille N*2) qui représente le produit de A*B

Produit de deux grands nombres

Notons :

- grand_nombre Produit (grand_nombre A, B)
  - Une fonction qui prend en entrée deux grands nombres A et B de taille N et retourne un grand nombre (de taille N*2) qui représente le produit de A*B
- grand_nombre addition (grand_nombre A, B)
  - Une fonction qui prend en entrée deux grands nombres A et B de taille N et retourne un grand nombre (de taille N) qui représente le produit de A+B

Produit de deux grands nombres

Notons :

- grand_nombre Produit (grand_nombre A, B)
  - Une fonction qui prend en entrée deux grands nombres A et B de taille N et retourne un grand nombre (de taille N*2) qui représente le produit de A*B
- grand_nombre addition (grand_nombre A, B)
  - Une fonction qui prend en entrée deux grands nombres A et B de taille N et retourne un grand nombre (de taille N) qui représente le produit de A+B
- grand_nombre soustraction (grand_nombre A, B)
  - Une fonction qui prend en entrée deux grands nombres A et B de taille N et retourne un grand nombre (de taille N) qui représente le produit de A-B

Produit de deux grands nombres

Notons :

- grand_nombre Produit (grand_nombre A, B)
  - Une fonction qui prend en entrée deux grands nombres A et B de taille N et retourne un grand nombre (de taille N*2) qui représente le produit de A*B
- grand_nombre addition (grand_nombre A, B)
  - Une fonction qui prend en entrée deux grands nombres A et B de taille N et retourne un grand nombre (de taille N) qui représente le produit de A+B
- grand_nombre soustraction (grand_nombre A, B)
  - Une fonction qui prend en entrée deux grands nombres A et B de taille N et retourne un grand nombre (de taille N) qui représente le produit de A-B
- grand_nombre insérer_zéro (int M, grand_nombre A)
  - Une fonction qui prend en entrée un grand nombre A de taille N et retourne un grand nombre (de taille N+M) en insérant m zéros

Grandes lignes de l'algorithme

```
grand_nombre Produit (int N, grand_nombre A, B)
{
grand_nombre p1,p2,p3,e,f;
p1=Produit(A_s, B_s);
p2 =Produit(A_i, B_i);
e=Addition(A_s, A_i);
f=Addition(B_s, B_i);
p3 = Produit(e,f);
insérer_zéro(N,p1);
f=Addition(p1,p2);
e=Soustraction (Soustraction (p3, p1), p2);
insérer_zéro(N/2,e);
retourner (Addition(e,f);
}
```

Complexité

```
/* Traitement des cas de base */ grand_nombre Produit (int N,
grand_nombre A, B)
{
grand_nombre p1,p2,p3,e,f;
p1=Produit(N/2, A_s, B_s);
p2 =Produit(N/2, A_i, B_i);
e=Addition(A_s, A_i);
f=Addition(B_s, B_i);
p3 = Produit(N/2, e,f);
insérer_zéro(N,p1);
f=Addition(p1,p2);
e=Soustraction (Soustraction (p3, p1), p2);
insérer_zéro(N/2,e);
retourner (Addition(e,f);
}
```

Complexité

/* Traitement des cas de base */ grand_nombre Produit (int N, grand_nombre A, B)
{
grand_nombre p1,p2,p3,e,f;
p1=Produit(N/2, A_s, B_s);
p2 =Produit(N/2, A_i, B_i);
e=Addition(A_s, A_i);
f=Addition(B_s, B_i);
p3 = Produit(N/2, e,f);
insérer_zéro(N,p1);
f=Addition(p1,p2);
e=Soustraction (Soustraction (p3, p1), p2);
insérer_zéro(N/2,e);
retourner (Addition(e,f);
}
Nous avons:

$$T(n) = 3 * T\left(\frac{N}{2}\right) + 8 * n$$

Complexité

$$T(n) = 3 * T(n/2) + 8 * n$$

Complexité

$$
\begin{aligned}
T(n) &= 3 * T(n/2) + 8 * n \\
&= 3 * \left(3 * T\left(\frac{n}{2^2}\right) + 8 * \left(\frac{n}{2}\right)\right) + 8 * n
\end{aligned}
$$

Complexité

$$
\begin{aligned}
T(n) &= 3 * T(n/2) + 8 * n \\
&= 3 * (3 * T\left(\frac{n}{2^2}\right) + 8 * \left(\frac{n}{2}\right)) + 8 * n \\
&= 3^2 * T\left(\frac{n}{2^2}\right) + (1 + \frac{3}{2}) * 8 * n
\end{aligned}
$$

Complexité

$$
\begin{aligned}
T(n) &= 3 * T(n/2) + 8 * n \\
&= 3 * (3 * T\left(\frac{n}{2^2}\right) + 8 * \left(\frac{n}{2}\right)) + 8 * n \\
&= 3^2 * T\left(\frac{n}{2^2}\right) + (1 + \frac{3}{2}) * 8 * n \\
&= 3^2 * (3 * T\left(\frac{n}{2^3}\right) + 8 * \left(\frac{n}{2^2}\right))) + (1 + \frac{3}{2}) * 8 * n
\end{aligned}
$$

Complexité

$$
\begin{aligned}
T(n) &= 3 * T(n/2) + 8 * n \\
&= 3 * (3 * T\left(\frac{n}{2^2}\right) + 8 * \left(\frac{n}{2}\right)) + 8 * n \\
&= 3^2 * T\left(\frac{n}{2^2}\right) + (1 + \frac{3}{2}) * 8 * n \\
&= 3^2 * (3 * T\left(\frac{n}{2^3}\right) + 8 * \left(\frac{n}{2^2}\right))) + (1 + \frac{3}{2}) * 8 * n \\
&= 3^3 * T\left(\frac{n}{2^3}\right) + (1 + \frac{3}{2} + \left(\frac{3}{2}\right)^2) * 8 * n
\end{aligned}
$$

Complexité

$$
\begin{aligned}
T(n) &= 3 * T(n/2) + 8 * n \\
&= 3 * (3 * T\left(\frac{n}{2^2}\right) + 8 * \left(\frac{n}{2}\right)) + 8 * n \\
&= 3^2 * T\left(\frac{n}{2^2}\right) + (1 + \frac{3}{2}) * 8 * n \\
&= 3^2 * (3 * T\left(\frac{n}{2^3}\right) + 8 * \left(\frac{n}{2^2}\right))) + (1 + \frac{3}{2}) * 8 * n \\
&= 3^3 * T\left(\frac{n}{2^3}\right) + (1 + \frac{3}{2} + \left(\frac{3}{2}\right)^2) * 8 * n \\
&. \\
&.
\end{aligned}
$$

Complexité

$$
\begin{aligned}
T(n) &= 3 * T(n/2) + 8 * n \\
&= 3 * (3 * T\left(\frac{n}{2^2}\right) + 8 * \left(\frac{n}{2}\right)) + 8 * n \\
&= 3^2 * T\left(\frac{n}{2^2}\right) + (1 + \frac{3}{2}) * 8 * n \\
&= 3^2 * (3 * T\left(\frac{n}{2^3}\right) + 8 * \left(\frac{n}{2^2}\right))) + (1 + \frac{3}{2}) * 8 * n \\
&= 3^3 * T\left(\frac{n}{2^3}\right) + (1 + \frac{3}{2} + \left(\frac{3}{2}\right)^2) * 8 * n \\
&\cdot \\
&\cdot \\
&= 3^m * T\left(\frac{n}{2^m}\right) + (1 + \frac{3}{2} + \left(\frac{3}{2}\right)^2 + \ldots + \left(\frac{3}{2}\right)^{m-1}) * 8 * n
\end{aligned}
$$

Complexité

$$
\begin{aligned}
T(n) &= 3 * T(n/2) + 8 * n \\
&= 3 * (3 * T\left(\frac{n}{2^2}\right) + 8 * \left(\frac{n}{2}\right)) + 8 * n \\
&= 3^2 * T\left(\frac{n}{2^2}\right) + (1 + \frac{3}{2}) * 8 * n \\
&= 3^2 * (3 * T\left(\frac{n}{2^3}\right) + 8 * \left(\frac{n}{2^2}\right))) + (1 + \frac{3}{2}) * 8 * n \\
&= 3^3 * T\left(\frac{n}{2^3}\right) + (1 + \frac{3}{2} + \left(\frac{3}{2}\right)^2) * 8 * n \\
&\cdot \\
&\cdot \\
&= 3^m * T\left(\frac{n}{2^m}\right) + (1 + \frac{3}{2} + \left(\frac{3}{2}\right)^2 + \ldots + \left(\frac{3}{2}\right)^{m-1}) * 8 * n
\end{aligned}
$$

Complexité

L'expression :

$$(1 + \frac{3}{2} + \left(\frac{3}{2}\right)^2 + ... + \left(\frac{3}{2}\right)^{m-1})$$

est une suite géométrique, qui est égale à :

$$\frac{\left(\frac{3}{2}\right)^m - 1}{\frac{3}{2} - 1}$$

qui est aussi égale à :

$$2 * \left(\frac{3}{2}\right)^m - 2$$

Complexité

$$T(n) = 3^m * T\left(\frac{n}{2^m}\right) + \left(1 + \frac{3}{2} + \left(\frac{3}{2}\right)^2 + \ldots + \left(\frac{3}{2}\right)^{m-1}\right) * 8 * n$$

Complexité

$$T(n) = 3^m * T\left(\frac{n}{2^m}\right) + \left(1 + \frac{3}{2} + \left(\frac{3}{2}\right)^2 + \ldots + \left(\frac{3}{2}\right)^{m-1}\right) * 8 * n$$

Complexité

$$
\begin{aligned}
T(n) &= 3^m * T\left(\frac{n}{2^m}\right) + \left(1 + \frac{3}{2} + \left(\frac{3}{2}\right)^2 + \ldots + \left(\frac{3}{2}\right)^{m-1}\right) * 8 * n \\
&= 3^m * T\left(\frac{n}{2^m}\right) + \left(2 * \left(\frac{3}{2}\right)^m - 2\right) * 8 * n
\end{aligned}
$$

Complexité

$$
\begin{aligned}
T(n) &= 3^m * T\left(\frac{n}{2^m}\right) + \left(1 + \frac{3}{2} + \left(\frac{3}{2}\right)^2 + \ldots + \left(\frac{3}{2}\right)^{m-1}\right) * 8 * n \\
&= 3^m * T\left(\frac{n}{2^m}\right) + \left(2 * \left(\frac{3}{2}\right)^m - 2\right) * 8 * n
\end{aligned}
$$

Question

Que vaut m?

Réponse

m est tel que : $T\left(\frac{n}{2^m}\right) = T(1)$, c'est à dire :

$$\frac{n}{2^m} = 1.$$

C'est-à-dire : $n = 2^m$.
Appliquons le *log* base (2), ce qui donne:

$$m = log_2(n).$$

On sait aussi que :

$$m = log_2(n) = \frac{log_3(n)}{log_3(2)}$$

Complexité

$$T(n) = 3^m * T\left(\frac{n}{2^m}\right) + \left(2 * \left(\frac{3}{2}\right)^m - 2\right) * 8 * n$$

Complexité

$$
\begin{aligned}
T(n) &= 3^m * T\left(\frac{n}{2^m}\right) + \left(2 * \left(\frac{3}{2}\right)^m - 2\right) * 8 * n \\
&= 3^m * T\left(\frac{n}{2^m}\right) + \left(2 * \left(\frac{3}{2}\right)^m - 2\right) * 8 * n
\end{aligned}
$$

Complexité

$$
\begin{aligned}
T(n) &= 3^m * T\left(\frac{n}{2^m}\right) + \left(2 * \left(\frac{3}{2}\right)^m - 2\right) * 8 * n \\
&= 3^m * T\left(\frac{n}{2^m}\right) + \left(2 * \left(\frac{3}{2}\right)^m - 2\right) * 8 * n \\
&= 3^{\frac{log_3(n)}{log_3(2)}} + \left(2 * \left(\frac{3}{2}\right)^{\frac{log_3(n)}{log_3(2)}} - 2\right) * 8 * n
\end{aligned}
$$

Complexité

$$
\begin{aligned}
T(n) &= 3^m * T\left(\frac{n}{2^m}\right) + \left(2 * \left(\frac{3}{2}\right)^m - 2\right) * 8 * n \\
&= 3^m * T\left(\frac{n}{2^m}\right) + \left(2 * \left(\frac{3}{2}\right)^m - 2\right) * 8 * n \\
&= 3^{\frac{log_3(n)}{log_3(2)}} + \left(2 * \left(\frac{3}{2}\right)^{\frac{log_3(n)}{log_3(2)}} - 2\right) * 8 * n
\end{aligned}
$$

Complexité

Rappel :

$$3^{\frac{log_3(n)}{log_3(2)}} = \left(3^{log_3(n)}\right)^{\frac{1}{log_3(2)}}$$

Complexité

Rappel :

$$
\begin{aligned}
3^{\frac{log_3(n)}{log_3(2)}} &= \left(3^{log_3(n)}\right)^{\frac{1}{log_3(2)}} \\
&= n^{\frac{1}{log_3(2)}}
\end{aligned}
$$

Complexité

Rappel :

$$
\begin{aligned}
3^{\frac{log_3(n)}{log_3(2)}} &= \left(3^{log_3(n)}\right)^{\frac{1}{log_3(2)}} \\
&= n^{\frac{1}{log_3(2)}} \\
&= n^{1.584}
\end{aligned}
$$

Complexité

Rappel :

$$
\begin{aligned}
3^{\frac{log_3(n)}{log_3(2)}} &= \left(3^{log_3(n)}\right)^{\frac{1}{log_3(2)}} \\
&= n^{\frac{1}{log_3(2)}} \\
&= n^{1.584}
\end{aligned}
$$

Complexité

$$T(n) = 3^{\frac{log_3(n)}{log_3(2)}} + \left(2 * \left(\frac{3}{2}\right)^{\frac{log_3(n)}{log_3(2)}} - 2\right) * 8 * n$$

Complexité

$$
\begin{aligned}
T(n) &= 3^{\frac{log_3(n)}{log_3(2)}} + \left(2 * \left(\frac{3}{2}\right)^{\frac{log_3(n)}{log_3(2)}} - 2\right) * 8 * n \\
&= n^{1.584} + \left(2 * \left(\frac{3}{2}\right)^{\frac{log_3(n)}{log_3(2)}} - 2\right) * 8 * n
\end{aligned}
$$

Complexité

$$
\begin{aligned}
T(n) &= 3^{\frac{log_3(n)}{log_3(2)}} + \left(2 * \left(\frac{3}{2}\right)^{\frac{log_3(n)}{log_3(2)}} - 2\right) * 8 * n \\
&= n^{1.584} + \left(2 * \left(\frac{3}{2}\right)^{\frac{log_3(n)}{log_3(2)}} - 2\right) * 8 * n
\end{aligned}
$$

Complexité

Rappel :

Complexité

Rappel :

$$\left(\frac{3}{2}\right)^{\frac{log_3(n)}{log_3(2)}} = \left(\frac{3^{\frac{log_3(n)}{log_3(2)}}}{2^{log_2(n)}}\right) = \left(\frac{3^{\frac{log_3(n)}{log_3(2)}}}{n}\right)$$

Complexité

Rappel :

$$\left(\frac{3}{2}\right)^{\frac{log_3(n)}{log_3(2)}} = \left(\frac{3^{\frac{log_3(n)}{log_3(2)}}}{2^{log_2(n)}}\right) = \left(\frac{3^{\frac{log_3(n)}{log_3(2)}}}{n}\right) = \left(\frac{n^{1.584}}{n}\right)$$

Complexité

Rappel :

$$\left(\frac{3}{2}\right)^{\frac{log_3(n)}{log_3(2)}} = \left(\frac{3^{\frac{log_3(n)}{log_3(2)}}}{2^{log_2(n)}}\right) = \left(\frac{3^{\frac{log_3(n)}{log_3(2)}}}{n}\right) = \left(\frac{n^{1.584}}{n}\right)$$

Complexité

$$T(n) = 3^{\frac{log_3(n)}{log_3(2)}} + \left(2 * \left(\frac{3}{2}\right)^{\frac{log_3(n)}{log_3(2)}} - 2\right) * 8 * n$$

Complexité

$$
\begin{aligned}
T(n) &= 3^{\frac{log_3(n)}{log_3(2)}} + \left(2 * \left(\frac{3}{2}\right)^{\frac{log_3(n)}{log_3(2)}} - 2\right) * 8 * n \\
&= n^{1.584} + \left(2 * \left(\frac{n^{1.584}}{n}\right) - 2\right) * 8 * n
\end{aligned}
$$

Complexité

$$
\begin{aligned}
T(n) &= 3^{\frac{log_3(n)}{log_3(2)}} + \left(2 * \left(\frac{3}{2}\right)^{\frac{log_3(n)}{log_3(2)}} - 2\right) * 8 * n \\
&= n^{1.584} + \left(2 * \left(\frac{n^{1.584}}{n}\right) - 2\right) * 8 * n
\end{aligned}
$$

Complexité

$$
\begin{aligned}
T(n) &= 3^{\frac{log_3(n)}{log_3(2)}} + \left(2 * \left(\frac{3}{2}\right)^{\frac{log_3(n)}{log_3(2)}} - 2\right) * 8 * n \\
&= n^{1.584} + \left(2 * \left(\frac{n^{1.584}}{n}\right) - 2\right) * 8 * n \\
&= n^{1.584} + 2 * 8 * n^{1.584} - 2 * 8 * n
\end{aligned}
$$

Complexité

$$
\begin{aligned}
T(n) &= 3^{\frac{log_3(n)}{log_3(2)}} + \left(2 * \left(\frac{3}{2}\right)^{\frac{log_3(n)}{log_3(2)}} - 2\right) * 8 * n \\
&= n^{1.584} + \left(2 * \left(\frac{n^{1.584}}{n}\right) - 2\right) * 8 * n \\
&= n^{1.584} + 2 * 8 * n^{1.584} - 2 * 8 * n
\end{aligned}
$$

Complexité

$$
\begin{aligned}
T(n) &= 3^{\frac{log_3(n)}{log_3(2)}} + \left(2 * \left(\frac{3}{2}\right)^{\frac{log_3(n)}{log_3(2)}} - 2\right) * 8 * n \\
&= n^{1.584} + \left(2 * \left(\frac{n^{1.584}}{n}\right) - 2\right) * 8 * n \\
&= n^{1.584} + 2 * 8 * n^{1.584} - 2 * 8 * n
\end{aligned}
$$

Complexité

$$
\begin{aligned}
T(n) &= 3^{\frac{log_3(n)}{log_3(2)}} + \left(2 * \left(\frac{3}{2}\right)^{\frac{log_3(n)}{log_3(2)}} - 2\right) * 8 * n \\
&= n^{1.584} + \left(2 * \left(\frac{n^{1.584}}{n}\right) - 2\right) * 8 * n \\
&= n^{1.584} + 2 * 8 * n^{1.584} - 2 * 8 * n \\
&\in O(n^{1.584})
\end{aligned}
$$

Complexité

$$
\begin{aligned}
T(n) &= 3^{\frac{log_3(n)}{log_3(2)}} + \left(2 * \left(\frac{3}{2}\right)^{\frac{log_3(n)}{log_3(2)}} - 2\right) * 8 * n \\
&= n^{1.584} + \left(2 * \left(\frac{n^{1.584}}{n}\right) - 2\right) * 8 * n \\
&= n^{1.584} + 2 * 8 * n^{1.584} - 2 * 8 * n \\
&\in O(n^{1.584})
\end{aligned}
$$