Vendredi 09 Octobre 2020

CM2 (Suite):

Il y a 2 semaines, on avait étudié Fib1 et Fib2 pour calculer le terme n-ième de la suite de Fibonacci.

La semaine dernière, on avait étudié Fib3 qui repose sur le principe matriciel:

$$\begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^n \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

On veut concevoir cet algorithme par la méthode « diviser pour régner ».
On applique cette approche à la multiplication de deux entiers sur n bits :

$$X \times Y = ac\,2^{n} + (ad + bc)\,2^{n/2} + bd$$

où a et b sont les deux moitiés des bits de X
c et d ——————————— Y

On transcrit cette formule en appels récursifs.
Or multiplier en binaire par 2^n correspond à rajouter n 0 au nombre binaire résultant de la multiplication

On applique cette approche à la multiplication de deux entiers sur n bits :

$$X \times Y = ac\,2^{n} + (ad + bc)\,2^{n/2} + bd$$

où a et b sont les deux moitiées des bits de X
c et d ______________________________ Y

On transcrit cette formule en appels récursifs.
Or multiplier en binaire par 2^n correspond à rajouter n 0 au nombre binaire résultant de la multiplicatio

<u>Ce qui donne en décimal</u> : Il faut barrer les "bits" sur le polycopié et remplacer par des «digits».

Dans chaque appel récursif, on continue de diviser les arguments jusqu'à tomber sur le cas de base : un entier d'un seul digit.

Puis on remonte l'arbre des appels récursifs en appliquant à chaque fois la formule de recombinaison.

Pour résoudre ce problème de taille n, on résoud 4 problèmes de taille $\frac{n}{2}$, donc 16 problèmes de taille $\frac{n}{8}$ ... et ainsi de suite.

Donc on résoud 4^k sous-problèmes de taille $\frac{n}{2^k}$ selon la

Donc on résoud 4^k sous-problèmes de taille $\frac{n}{2^k}$ selon la hauteur de l'arbre, notée k. Les feuilles sont constituées des problèmes de taille 1.

On détermine le coût de chaque noeud $T\left(\frac{n}{2^k}\right)$ de l'arbre en examinant les opérations de la formule.
Lors de la phase de recombinaison, les calculs en chaque noeud coûtent $O\left(\frac{n}{2^k}\right)$ opérations élémentaires.

On somme sur tous les niveaux de l'arbre :

$$T(n) = \sum_{k=0}^{\log_2(n)} \underbrace{4^k}_{\text{nombre de problèmes}} \underbrace{\left(\frac{n}{2^k}\right)}_{\text{leur taille}}$$

noeud contient $O\left(\frac{n}{2^k}\right)$ opérations élémentaires.

On somme sur tous les niveaux de l'arbre :

$$T(n) = \sum_{k=0}^{\log_2(n)} \underbrace{4^k}_{\text{nombre de problèmes}} \underbrace{\left(\frac{n}{2^k}\right)}_{\text{leur taille}}$$

$= \dots$

$= n(2n-1)$

$= 2n^2 - 1$

On a une complexité en $O(n^2)$, l'algorithme Fib3 ne fait pas mieux que Eib2.

Quizz : Diviser pour régner

B est la bonne réponse.

On veut améliorer la complexité de notre méthode « diviser pour régner » :

- on peut améliorer la complexité de la combinaison des réponses de sous-problèmes ;
- diminuer le facteur de branchement, donc la hauteur de l'arbre.

l'intérêt de la 2ème solution : on veut un facteur

On va s'intéresser à la 2ème solution : on veut un facteur de branchement de 3 plutôt que 4.

On passe par un algorithme décrit dans le diapo.

<u>Question</u> sur les nombres complexes.

On peut calculer $ac - bd$ et $ad + bc$ avec seulement 3 multiplications : voir <u>La solution de Gauss</u>.

Les seuls produits sont P_3, P_4 et P_5. Toutes les autres opérations sont des additions.

L'astuce est de multiplier $(a+b)$ par $(c+d)$ $[P_3]$.

On peut calculer $ac - bd$ et $ad + bc$ avec seulement 3 multiplications : voir la solution de Gauss.
Les seuls produits sont P_3, P_4 et P_5. Toutes les autres opérations sont des additions.
L'astuce est de multiplier $(a+b)$ par $(c+d)$ $[P_3]$.

On a 3 produits et 5 additions.

On veut appliquer cette méthode à « diviser pour régner ». Il n'y a plus que 3 appels récursifs.
(Voir mult2 gaussifiée).

Dans l'arbre, on a 3^k problèmes de taille $\frac{n}{2^k}$ à chaque étage k de l'arbre.

$$T(n) = \sum_{k=0}^{\log_2(n)} 3^k \frac{n}{2^k} = \sum_{k=0}^{\log_2(n)} n \left(\frac{3}{2}\right)^k$$

$$= \dots$$

$$= 3n^{\log_2(3)} - 2n$$

La complexité obtenue est $O(n^{1,58})$ ce qui est bien mieux que $O(n^2)$ si l'on compare les courbes des 2 fonctions.

Ce qui compte dans la méthode diviser pour régner est :
- le nombre de problèmes
- la taille de ces problèmes
- la complexité de la fusion

⚠ **Le théorème maître** : (Voir diapo du même nom)

Ce théorème peut s'écrire avec des complexités en O ou en Θ.

Il permet de calculer immédiatement sans suivre les appels récursifs dans une méthode « diviser pour régner ».

La preuve est liée à l'arbre des appels récursifs : même analyse que vue précédemment pour mult et mult2.

⚠ Le théorème maître : (Voir diapo du même nom)

Ce théorème peut s'écrire avec des complexités en O ou en Θ.

Il permet de calculer immédiatement sans suivre les appels récursifs dans une méthode « diviser pour régner ».

La preuve est liée à l'arbre des appels récursifs : même analyse que vue précédemment pour mult et mult2.

Quizz : Théorème maître

B On ne peut pas appliquer le théorème maître car les problèmes ne sont pas de même taille : $\frac{n}{3} \neq \frac{n}{2}$.

Tous les sous-problèmes doivent être de même taille.

On l'applique à mult, mult2 et TRI_FUSION pour illustrer son fonctionnement où :

- a = nombre d'appels récursif
- b = taille des sous problèmes
- d = exposant de la complexité linéaire

On retrouve bien nos complexités précédentes, mais bien plus facilement.

[illegible] : Diviser pour régner :

On retrouve bien nos complexités précédentes, mais bien plus facilement.

Quizz : Diviser pour régner.

(A faire soi-même pour s'entraîner)

$a = 1$, $b = 2$ et $d = 0$

On a $1 = 2^0 \iff a = b^d \iff d = \log_b a$

donc $T(n) \in \Theta(\log n)$

A est la bonne réponse.

Cette formule précise de complexité correspond à la recherche dichotomique dans un tableau trié.

$a = 1$, $b = 2$ et $d = 0$

On a $1 = 2^0 \iff a = b^d \iff d = \log_b a$

donc $T(n) \in \Theta(\log n)$

A est la bonne réponse.

Cette formule précise de complexité correspond à la recherche dichotomique dans un tableau trié.

- Autre exemple (Paire de points les plus proches)

On veut déterminer la paire de points les plus proches dans un plan, ce le plus efficacement possible.

On fait la méthode diviser pour régner.

Division : couper le plan en 2

Régner : trouver la paire la plus proche dans chaque section (appels récursifs)

P ⚠ Envisager les paires avec un point dans chaque sous-région.

Combiner : Retourner la meilleure solution possible

Pour résoudre P : on définit une bande centrale :

Combiner : Retourner la meilleure solution possible

Pour résoudre P : on définit une bande centrale :

"On trie les points selon leurs ordonnées puis on les compare en remodelant l'espace (voir la diapo avec la propriété et la "Preuve (arguments)").

2 points ne peuvent être dans un même carré, au plus 1 point par carré.
Au delà de deux carrés de séparation, on ne s'intéresse pas à la paire.
Pour calculer la plus petite distance, on compare les points aux autres $\Rightarrow$ Complexité de $\boxed{O(n)}$.

avec la propriété et la "Preuve (arguments)").

2 points ne peuvent être dans un même carré, au plus 1 point par carré.
Au delà de deux carrés de séparation, on ne s'intéresse pas à la paire.
Pour calculer la plus petite distance, on compare les points aux autres $\Rightarrow$ Complexité de $\boxed{O(n)}$.

En applicant le théorème maître, on a $T(n) \in \boxed{O(n \log n)}$

Cours 3 : Introduction au graphe :

Un cycle parcourant un graphe en empruntant une unique fois chaque arête est appelé un cycle eulérien.

Chaque point d'un graphe se nomme un sommet. Les lignes les reliant sont des arêtes.

Souvent, on veut trouver | la plus courte chaîne / le plus court chemin | pour aller d'un sommet à un autre : le choix d'un itinéraire.

Souvent, on veut trouver | la plus courte chaîne | le plus court chemin | pour aller d'un sommet à un autre : le choix d'un itinéraire.

Graphe non-orienté = les arêtes n'ont pas de direction
Graphe orienté = les arcs sont orientées.

Dans l'exemple Emploi du temps, les arêtes représentent l'incompatibilité. La couleur d'un sommet correspond à un créneau du cours : 2 sommets voisins sont de couleur différente. On veut utiliser le nombre minimum de couleurs, ici 3 : pas moins, car il y a un triangle (sous-graphe complet) !

Le nombre d'heures minimum = nombre chromatique

Le nombre d'heures minimum = nombre chromatique

1. Graphe orienté (Voir diapo)

Le graphe peut être écrit sous forme de liste, le dessin sert de support visuel.

Degré : nombre de sommets incidents à un sommet précis.

2. Graphe non orienté (voir diapo)

Quizz : Degré des sommets

Chaque arête a deux extrémités.
D est la bonne réponse. On compte le nombre d'extrémités d'arêtes

Quizz : Suite Graphique

On peut s'aider de la réponse à la question précédente : la somme des degrés doit forcément être paire ($2m$).

C La suite 1 et la suite 3 sont graphiques.

c) La suite 1 et la suite 3 sont graphiques.

On peut tracer la suite 3 pour s'en convaincre.

3. Graphe partiel (voir diapo)

On supprime certains arcs d'un graphe mais on garde les sommets

4. Sous-graphe induit (voir diapo)

On sélectionne des sommets et on garde tous les arcs qui les concernent.

On sélectionne des sommets et on garde tous les arcs qui les concernent.

Quizz : Graphe partiel et sous-graphe

1. A On a gardé tous les sommets, mais on a supprimé des arcs.
2. C On a extrait un sous-ensemble de sommets ET on a supprimé certains des arcs les reliant.

On les représente avec des matrices ~~ou des graphes~~ : les valeurs booléennes indiquent si un arc existe entre deux sommets (On part des sommets en ligne pour arriver au sommet en colonnes dans le cas d'un graphe orienté).

ou avec un tableau contenant des listes chaînées de successeurs.