

Structures de données et algorithmes

Répétition 8: Résolution de problèmes

Jean-Michel BEGON

4 mai 2018

1 Multiplication de polynômes

On représente un polynôme par un tableau de ses coefficients par ordre croissant de degré. Par exemple, le polynôme $5 + 10x^2 + 6x^3$ est représenté par le tableau $[5, 0, 10, 6]$. Un algorithme simple pour multiplier deux polynômes est le suivant :

```
MULTIPLY( $A, B$ )
1   $n = A.length$ 
2   $m = B.length$ 
3  Soit  $P$  un tableau de taille  $n + m - 1$ 
4  for  $i = 1$  to  $P.length$ 
5       $P[i] = 0$ 
6  for  $i = 1$  to  $n$ 
7      for  $j = 1$  to  $m$ 
8           $P[i + j] = P[i + j] + A[i] + B[j]$ 
9  return  $P$ 
```

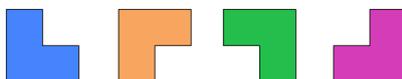
- Déterminez la complexité de en fonction de n et m , les degrés des deux polynômes.
- Proposez un algorithme de multiplication de type diviser-pour-régner. Vous pouvez supposer que les deux polynômes sont de degrés $n = 2^k$.
- Analysez la complexité de votre algorithme.

2 Multiplication matricielle

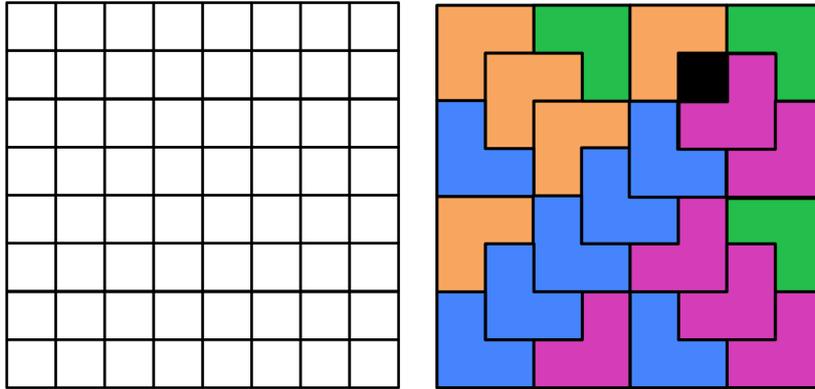
Proposez un algorithme diviser-pour-régner afin de multiplier deux matrices $n \times n$ ($n = 2^k, k \geq 0$). Quelle est la complexité de la solution ?

3 Carré de triminos

On souhaite recouvrir un carré de taille $N \times N$ ($N = 2^m, m \geq 0$) à l'aide des quatre types triminos, à l'exception d'une case vide dont les coordonnées sont données.



Les triminos de base



Carré 8x8 vide et recouvert

Proposez un algorithme pour résoudre ce problème.

4 Distance minimum

Soit un ensemble $P = \{p_1, p_2, \dots, p_n\}$ de points ($n = 2^k$, $k > 0$) répartis dans un plan et tels que $p_i = (x_i, y_i)$. On cherche un algorithme qui détermine la distance (euclidienne) entre les deux points les plus proches.

1. Proposez une approche exhaustive pour solutionner ce problème.
2. Proposez un algorithme diviser-pour-régner pour ce problème.
3. Comparez la complexité des deux algorithmes.

5 Approximation gloutonne

Proposez une approximation gloutonne, ainsi qu'un contre-exemple démontrant sa sous-optimalité pour les problèmes suivants :

1. La chasse au trésor.
2. Découpage de la tige d'acier.
3. L'ordre des multiplication de matrices.

6 La course de Julien (adapté de CLRS, 16.2-4)

Julien participe à une course à pieds qui consiste à relier une ville A à une ville B . Etant donné la contenance de son bidon, il sait qu'il peut parcourir m mètres sans tomber à cours d'eau. Il a également pu obtenir des organisateurs du marathon la liste des endroits où il pourra remplir son bidon et les distances entre ces endroits.

Julien aimerait minimiser le nombre d'arrêts qu'il devra effectuer lors de sa course pour remplir son bidon. Donnez une méthode efficace pour déterminer quels arrêts il devrait faire. Montrez que votre stratégie donne une solution optimale et discutez sa complexité.

7 L'ordonnement

L'ordonnement des tâches est un problème fréquent en informatique : on dispose d'un ensemble $S = \{s_1, s_2, \dots, s_n\}$ de n tâches qu'on doit ordonner. A chaque tâche s_i est associé un

temps d'exécution t_i . Une fois l'ordre des tâches fixé, on peut également associer à chaque tâche un temps de complétion (ou temps de réponse), c'est-à-dire, le temps qui s'est écoulé jusqu'à la fin de la tâche.

Par exemple, si on dispose des tâches s_1 et s_2 dont les temps d'exécution respectifs sont $t_1 = 3$ et $t_2 = 5$ et si on exécute les tâches dans l'ordre $\langle s_1, s_2 \rangle$, alors les temps de complétion seront :

- $C_1 = t_1 = 3$
- $C_2 = C_1 + t_2 = 3 + 5 = 8$

En outre, la somme des temps de complétion sera $C_\Sigma = \sum_i C_i = C_1 + C_2 = 11$.

On souhaite disposer d'un algorithme d'ordonnancement des tâches qui minimise la somme des temps de complétion (C_Σ).

1. Quelle serait la complexité d'un algorithme *brute-force* pour résoudre ce problème ?
2. Est-il possible de faire mieux ? (*i.e.* le problème dispose-t-il de la propriété de sous-structure optimale ?)
3. Le cas échéant, proposez un algorithme efficace pour résoudre ce problème.

8 Intervalles unitaires (CLRS, 16.2-5)

Etant donné un ensemble x_1, x_2, \dots, x_n de points sur l'axe réel, donner un algorithme efficace pour déterminer le plus petit ensemble d'intervalles fermés de longueur unitaire qui contient tous les points. Montrez que votre algorithme est correct.

9 Je vais au marché

Vous devez acheter n produits. Le prix initial de chaque produit est 1 euro mais chaque jour le prix du i ème produit augmente d'un facteur $f(i)$. Développez un algorithme efficace pour trouver l'ordre optimal dans lequel les produits devraient être achetés, en supposant que vous pouvez acheter au plus un produit chaque jour.

Bonus — La peinture (adapté de l'IEEEExtreme 2016)

Vous souhaitez mettre en couleur une mosaïque. Comme vous êtes méthodiques, vous avez déjà acheté tous les pigments nécessaires et vous avez décidé de peindre la toile de haut en bas et droite à gauche. Vous connaissez donc exactement la suite de couleurs à utiliser. Vous disposez de deux pinceaux et souhaitez minimiser le nombre de fois que vous devez nettoyer votre pinceau. Quelle stratégie allez-vous adapter ?