

TD n°7

TD Noté

Ce TD noté est à réaliser individuellement. Toute communication est interdite. L'épreuve est composée de deux questions et dure deux heures. La seule documentation autorisée est une feuille A4 manuscrite. Toute autre documentation ou appareil électronique est interdit. L'utilisation de téléphones portables est interdit, même comme montre.

Algorithmes diviser-pour-régner

Rappel : Master Theorem

Vous pouvez utiliser le Master Theorem ci-dessous pour analyser la complexité d'algorithmes récursifs. Soit une fonction T qui représente la complexité d'un algorithme sur une donnée de taille n de la forme

$$T(n) = aT\left(\frac{n}{b}\right) + f(n).$$

Dans chacun des cas décrits ci-après, la fonction T est de l'ordre de grandeur suivant :

Cas “ f petit” : Si $f(n) = 0$ ou, plus généralement, $f(n) = O(n^{\log_b(a)-\epsilon})$ alors $T(n) = \Theta(n^{\log_b(a)})$

Cas “ f grand” : Si $f(n) = \Omega(n^{\log_b(a)+\epsilon})$ et si $af\left(\frac{n}{b}\right) \leq cf(n)$ avec $c < 1$ alors $T(n) = \Theta(f(n))$

Cas intermédiaire : Si $f(n) = \Theta(n^{\log_b(a)})$ alors $T(n) = \Theta(n^{\log_b(a)} \log n)$

Fréquence des éléments d'un tableau

Exercice 1 Soit $T[1 \dots n]$ un vecteur d'entiers non nécessairement distincts. Pour chaque valeur u dans le tableau, n_u désigne le nombre d'occurrences de u dans T . Proposez un algorithme de type diviser-pour-régner qui retourne la liste $\{(u, n_u)\}$ pour chaque u élément du tableau. Exemple : soit $T[1 \dots 6] = [12, 3, 12, 4, 3, 12]$. La liste solution est alors $(12, 3), (3, 2), (4, 1)$ signifiant : 12 apparaît 3 fois, 3 apparaît 2 fois, et 4 apparaît 1 fois. Donnez la récurrence qui compte la complexité de votre algorithme, et résolvez la récurrence.

Exercice 2 Modifiez l'algorithme précédent pour qu'il retourne la liste $\{(u, n_u)\}$ en ordre croissant de u . Donnez la récurrence qui compte la complexité de votre algorithme, et résolvez la récurrence.

Algorithmes de programmation dynamique

Exercice 3 Le kayak est un sport nautique qui se pratique sur une embarcation appelée kayak, et la navigation se fait sur une rivière dans le sens du courant (on ne peut pas naviguer dans le sens contraire du courant.) Il y a N stations de location de kayak le long d'une rivière. Les stations se font concurrence et pratiquent des tarifs variables. Une table $\text{tarif}[i,j]$ donne le tarif de location d'un kayak pris à la station i pour se rendre jusqu'à la station j . En pratique, il peut être plus économique, pour aller de la station i à la station j , de faire plusieurs arrêts entre i et j et louer sur les segments intermédiaires.

1. Donnez une récurrence qui permet de calculer le coût optimal de location pour aller de la station i à la station j .

2. Donnez les dimensions du tableau que vous utiliserez pour faire le calcul du coût optimal, et expliquez comment celui-ci doit être rempli pour donner la valeur de la solution optimale.
3. Donnez l'algorithme de type programmation dynamique qui calcule la valeur de la solution optimale.
4. Donnez la complexité en temps et en espace de votre algorithme.