

TD n°7

TD Noté

Ce TD note est a realiser individuellement. Toute communication est interdite. L'epreuve est composee de deux questions et dure deux heures. La seule documentation autorisee est une feuille A4 manuscrite. Toute autre documentation ou appareil electronique est interdit. L'utilisation de telephones portables est interdit, même comme montre.

Algorithmes diviser-pour-régner

Rappel : Master Theorem

Vous pouvez utiliser le Master Theorem ci-dessous pour analyser la complexite d'algorithmes recursifs. Soit une fonction T qui represente la complexite d'un algorithme sur une donnee de taille n de la forme

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

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

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

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

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

Fréquence des éléments d'un tableau

Exercice 1

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.