

Compte rendu ProjetCorrection

IE-4B zhile zhang

Question 1 :

La fonction *corriger*, pour n'importe la taille du Texte, la boucle parcourera tous les éléments(mots) du tableau (Texte) jusqu'à la fin, on est toujours dans pire cas. Mais quand chaque mot(élément) parcouru, on appellera la fonction *recherche*, dans la fonction recherche, le pire cas est quand le mot à cherché n'est pas dans le dictionnaire.

- a) La complexité de la fonction *recherche(dico, mot)* est de $O(\text{dico.length}) = O(N)$ car chaque élément du tableau est parcouru au plus une fois.
- b) La complexité de la fonction *corriger(dico, texte)* est de $O(\text{texte.length} * \text{dico.length} + 1) = O(T * N)$ car chaque élément du tableau texte est appelé la fonction *recherche* au plus une fois .

Question 2 :

Le pire cas est tout le temps puisque la fonction divise le tableau en sous-tableaux de longueur 2 ou 1 et est obligée de parcourir chaque élément pour tri.

- a) La complexité de la fonction *fusion(String[] t1, String[] t2)* est de $O(t1.length + t2.length)$ car chaque élément des tableaux est parcouru au plus une fois.
- b) La complexité de la fonction *sousTableau(String[] t, int i, int j)* est de $O(j-i)$ car chaque élément du tableau t entre l'index de i à j est parcouru au plus une fois.
- c) La complexité de la fonction *triFusion(String[] t)* : $T(t.length) = 2 * T(2/t.length) + t.length$, c.à.d $T(N) = 2T(N/2) + N$, car dans la fonction *sousTableau*, quand $i=0$, $j=t.length/2$, $T(j-i) = T(N/2)$, et il y a deux sous-tableaux, et dans la fonction *fusion*, quand $t1.length = N/2$, $t2.length = N/2$, $T(t1.length + t2.length) = T(N/2 + N/2) = T(N)$. Pareil, $T(N/2) = 2T(N/4) + N/2$, $T(N) = 4T(N/4) + 2N$, car le fonction *triFusion* est une fonction récursive, on doit continuer diviser, tri, fusion, et calcule $T(N/4)$, $T(N/8)$, jusqu'à 2, on peut obtenir $T(N) = 2^{x-1} T(N/2^{x-1}) + (X-1) * N = 2^{x-1} T(2) + (X-1) * N$, X est le fois qu'on divise jusqu'à la longueur de chaque sous-tableau est 1, on obtient $N/2^{x-1} = 2$, donc $x = \log_2 N$, donc $T(N) = N * T(2) + N * \log_2 N$, et quand un tableau juste a 2 éléments, directement tri, alors $T(2) = 0$, donc $O(T(N)) = O(N * \log_2 N)$.

Question 3 :

Le pire cas est quand on recherche un élément(mot) qui n'est pas dans le tableau dico.

- a) La complexité de la fonction *rechercheDichoString(mot, dico)* est de :
 $O(\log_2(\text{dico.length}) + 1) = O(\log_2 N)$ car on note $\log_2(N) + 1$ est le nombre de fois que l'on peut diviser N par 2 jusqu'à atteindre 0. Et chaque division on cherche le mot dans une partie.
- b) La complexité de la fonction *corrigerRapide(text, dico)* est de :
 $O(\text{dico.length} * \log_2(\text{dico.length}) + \text{text.length} * \log_2(\text{dico.length}) + 1) = O(N * \log_2 N + T * \log_2 N)$ car d'abord on appelle la fonction *triFusion*, et chaque élément(mot) du tableau Text est appelé la fonction *rechercheDichoString* au plus une fois .
- c) La complexité de la fonction *corrigerDicoRapide(text)* est la complexité de la fonction *corrigerRapide(text, dico)*, donc $O(N * \log_2 N + T * \log_2 N)$.