

Portail Mathématiques, Informatique
Licence deuxième année (L2 Info, L2 maths, L2 MIASHS)

Algorithmique et structures de données
TD 9 – complexité

Exercice 1. Opérations sur les tableaux en C et les listes en python

Question 1. Donnez la complexité en moyenne pour les instructions suivantes sur un tableau en langage C :

- accéder à un élément du tableau à partir de son indice
- accéder à un élément du tableau à partir de la valeur de l'élément, considérez les deux cas : le tableau n'est pas trié et le tableau est trié
- supprimer un élément du tableau à partir de son indice
- insérer un élément dans le tableau à un indice

Question 2. En python, une liste n'est pas une liste chaînée mais ressemble plus à un tableau qui est géré dynamiquement (de la mémoire est allouée lorsque l'on ajoute des éléments).

On notera n la longueur d'une liste L . Nous aurons parfois besoin d'un second paramètre $k \leq n$ pour désigner la longueur d'une seconde liste L_2 .

D'après vous, quel est le coût des opérations suivantes ?

$L.append(x)$ $L.extend(L_2)$ $L.insert(i,x)$
 $x = L[i]$ $L[i] = 0$ $L.remove(x)$
 $x = \min(L)$ $x = \max(L)$ $L = []$
 $L_2 = L[i:i+k]$ $x \text{ in } L$

Exercice 2. Complexité des algorithmes de tris

Pour le calcul de la complexité en moyenne des algorithmes de tri, nous supposons que nous trions n entiers différents $E_n = \{e_1, \dots, e_n\}$ et que l'entrée est une permutation de E_n tirée avec la distribution uniforme (chaque permutation a la même probabilité d'être tirée).

Question 3. Rappelez comment est calculé le coût d'un algorithme de tri.
Est-ce que ce choix vous semble judicieux ?

Question 4. Redonnez l'algorithme du tri sélection et calculez son coût exact.

Notons $CS(n)$ la complexité en moyenne du tri sélection pour n entiers.

Calculez $CS(1), CS(2), CS(3), CS(4)$ et $CS(5)$.

Pourquoi n'y a-t-il pas de différence entre la complexité dans le pire des cas et la complexité en moyenne ?

Donnez la classe Θ de cet algorithme.

Question 5. Soient $n \in \mathbb{N}$ et t_1 le temps d'exécution du tri sélection pour n entiers.

Donnez t_2 le temps d'exécution du tri sélection pour $10n$ entiers.

En déduire le temps t_2 lorsque $t_1 = 1$ s, puis $t_1 = 1$ mn.

Question 6. Redonnez l'algorithme du tri insertion.

Notons $CI(n)$ la complexité en moyenne du tri insertion pour n entiers.

Calculez $CI(1), CI(2), CI(3), CI(4)$ et $CI(5)$.

Comparez le coût de l'algorithme dans le pire des cas et en moyenne.

Donnez la classe Θ de cet algorithme.

Question 7. Redonnez l'algorithme du tri fusion. Montrez que l'on a

$$CF(n) = F(n) + CF(\lfloor \frac{n}{2} \rfloor) + CF(\lceil \frac{n}{2} \rceil),$$

où $CF(n)$ désigne le coût du tri fusion pour n entiers et $F(n)$ désigne le coût de la fusion de deux tableaux de taille n_1 et n_2 tels que $n_1 + n_2 = n$. Montrez que l'on a $F(n) \leq n - 1$. Nous supposons pour les calculs que nous avons toujours $F(n) = n - 1$.
Calculez $CF(1), CF(2), CF(3), CF(4)$ et $CF(5)$.

Question 8. Nous allons maintenant calculer $CF(n)$ dans le cas où $n = 2^k$.

Calculez $CF(\lfloor \frac{n}{2} \rfloor)$ et $CF(\lceil \frac{n}{2} \rceil)$ en fonction de k . Que peut-on dire de la taille des sous-tableaux intervenant dans l'algorithme de tri fusion ?

Pour tout $i \in \{0, \dots, k\}$, posons $c_i = CF(2^i)$ et $f_i = F(2^i)$. Donnez c_0, c_1 et c_2 .

Montrez par récurrence que l'on a

$$c_i = (i - 1) 2^i + 1,$$

pour tout $i \in \{0, \dots, k\}$.

En déduire, $CF(n)$ pour $n = 2^k$. Quelle est la classe Θ du tri fusion ?

Question 9. * Comparez les complexités en moyenne des trois algorithmes de tri précédents pour $n \leq 5$.

On souhaite modifier l'algorithme de tri fusion, lors des appels récursifs. Lorsque nous avons un tableau d'au plus cinq entiers, on applique un des deux autres algorithmes. Suggérez-vous d'appliquer le tri sélection ou le tri insertion ? Justifiez votre réponse. Donnez l'arbre des appels récursifs pour $n = 45$ en précisant sur chaque nœud quelle fonction est appelée.

Timsort est un algorithme de tri hybride combinant le tri fusion et le tri insertion. C'est l'algorithme standard de tri utilisé par Python depuis la version 2.3.

Quelle est d'après-vous la classe de complexité de *Timsort* ?

Soit L une liste python, quelle est la différence entre les instructions $L.sort()$ et $sorted(L)$?

Exercice 3. Sélection des plus petits éléments d'un ensemble

On veut sélectionner les k plus petits entiers (non nécessairement triés) d'un ensemble de n entiers distincts (où $n > k$) donnés sous forme d'un tableau T non trié.

Question 10. Donnez un algorithme simple pour résoudre ce problème. Indication : inspirez-vous d'un algorithme de tri déjà étudié. Donnez la complexité dans le pire des cas et en moyenne de cet algorithme.

Question 11. * *QuickSelect* est un algorithme qui prend en entrée un tableau de n entiers non trié T et un entier $k \leq n$ et retourne le k ème plus petit élément de T .

Cet algorithme s'inspire de l'algorithme *QuickSort* et utilise la fonction *partition*. On prend le premier élément comme pivot. Soit t la position du pivot retournée par *partition*. Nous avons les trois cas suivants :

- a) $t = k - 1$, le pivot était l'élément recherché
- b) $t < k - 1$, on continue la recherche dans le sous-tableau contenant les éléments plus grands que le pivot
- c) $t > k - 1$, on continue la recherche dans le sous-tableau contenant les éléments plus petits que le pivot

Donnez la procédure *QuickSelect*.

Question 12. ** Adaptez *QuickSelect* pour obtenir les k plus petits éléments de T .