TD5 : Tableaux statiques

version imprimable

Exercice 1

On considère un ensemble de k données stockés dans un tableau de données statique T de N cases (avec $0 \leq k \leq N$).

Remarques :

- Les cases du tableau sont numérotées de 0 à N-1
- Les données sont de type quelconque mais chaque case ne peut contenir qu'une donnée
- Si i est un indice de case, T[i] désigne le contenu de la case
- N est fixé mais k varie en fonction du nombre de données stockées

Question 1

Propriété 1 : les données sont stockés dans les k premières cases du tableau. Ainsi, les cases de 0 à k-1 sont occupées et l'indice k désigne la première case libre.

Ecrire un algorithme permettant d'insérer une nouvelle donnée t dans le tableau T en conservant la propriété 1. Quelle est sa complexité?

Propriété 2 : il n'y a pas de doublons dans le tableau, autrement dit $\forall i,j < k$, si $i \neq j$ alors $T[i] \neq T[j]$.

Ecrire un algorithme permettant d'insérer une nouvelle donnée t dans le tableau T en conservant les propriétés 1 et 2. Quelle est sa complexité?

Propriété 3 : On suppose qu'il existe un ordre $<$ sur les données. $\forall i,j < k$, si $i < j$ alors $T[i] < T[j]$.

Écrire un algorithme permettant d'insérer une nouvelle donnée t dans le tableau T en conservant les propriétés 1, 2 et 3. Quelle est sa complexité?

Question 2

Un algorithme de *recherche* prend en argument une donnée t et retourne :

- Le numéro de la case contenant t si t est présent dans le tableau
- -1 sinon ("donnée absente")

Écrire un algorithme de recherche sachant que la propriétés 1 est respectée, un algorithme de recherche sachant que les propriétés 1 et 2 sont respectées, puis un algorithme de recherche sachant que les propriétés 1 et 2 et 3 sont respectées. Quelle est leur complexité?

Question 3

Un algorithme de *suppression* prend en argument une donnée `t` et :

- Supprime `t` du tableau si `t` est présent dans le tableau
- Ne fait rien sinon

Écrire un algorithme de suppression conservant la propriétés 1, un algorithme de suppression conservant les propriétés 1 et 2, puis un algorithme de suppression conservant les propriétés 1 et 2 et 3. Quelle est leur complexité?

Exercice 2

On appelle *liste* une structure abstraite ordonnée telle que l'on puisse accéder de manière directe à l'élément `i` et à laquelle on puisse ajouter (et supprimer) autant d'éléments que l'on souhaite. Une caractéristique importante de cette structure est son nombre d'éléments.

Une implémentation des listes peut être effectuée comme suit:

- On commence par créer un tableau de taille n = 1, le nombre initial d'éléments étant k = 0
- A chaque ajout d'élément:
  - si k < n,
    - ajouter l'élément à la position k
    - k ← k + 1
  - sinon :
    - allouer un tableau à 2 * n éléments et n ← n * 2
    - copier les k premiers éléments du tableau initial dans le nouveau tableau & supprimer le tableau initial.
    - ajouter l'élément à la position k
    - k ← k + 1

Montrez que la complexité de l'ajout de k éléments à la fin d'une liste originellement vide est O(k).

Exercice 3 (Suppression de doublons)

La structure de données utilisée ici est la liste. On considérera que la création d'une liste vide, l'ajout d'un élément en fin de liste & la lecture d'un élément dans une liste se font en O(1) opérations.

1. Donnez un algorithme permettant de résoudre le problème suivant :
   - Données : Une liste L & une valeur `val`.
   - Rendre : Une liste L2, restriction de L aux valeurs différentes de `val`.
   - Quelle est sa complexité ?
2. Utilisez la question précédente pour écrire un algorithme résolvant le problème suivant :
   - Données : Une liste L.
   - Rendre : Une liste L2 ne contenant qu'une seule occurrence de chaque valeur de L & en conservant le même ordre.
   - Quelle est sa complexité ?
3. Même question que précédemment, mais on considère que la liste L en entrée est triée. Donnez un algorithme en O(n) pour résoudre ce problème, où n est le nombre d''éléments de L.

4. Si l'ordre des éléments de L2 n'est pas important, proposez une meilleure solution à la question du point 2.

Exercice 4

On considère comme dans l'exercice 1 un tableau statique de données T de taille N. A la différence de l'exercice 1,

- Il est possible d'écrire plusieurs données dans une case de tableau :
  - chaque case de T peut contenir entre 0 et n données
  - $0 \leq k[i] \leq n$ est le nombre d'éléments contenus dans la case i
  - $0 \leq K \leq N$ est le nombre de cases effectivement occupées (la valeur de K peut varier au cours du temps).

1. En vous inspirant de l'exercice 2, montrez qu'il est possible d'insérer une donnée t dans une case i quelconque en temps constant
2. Si les données sont triées ($\forall i,j < K$, si $i < j$ alors $\forall\ t \in T[i], \forall\ s \in T[j], t < s$), montrez qu'il est possible d'insérer une donnée dans le tableau en O(log K).

From:
https://wiki.centrale-med.fr/informatique/ - **WiKi informatique**

Permanent link:
https://wiki.centrale-med.fr/informatique/tc_info:td2-2018-2019

Last update: **2019/07/31 11:01**