
TD n° 1 - Structures de données

Exercice 1.

Compléter le tableau en indiquant la complexité dans le pire cas pour chaque opération et chaque structure de donnée. L'ensemble des données noté S admet un ordre total, et x désigne un élément.

	liste non triée, simplement chaînée	liste triée, simplement chaînée	liste non triée doublement chaînée	liste triée, doublement chaînée	tableau dynamique non trié	tableau dynamique trié
RECHERCHER(x, S)						
INSÉRER(x, S)						
SUPPRIMER(x, S)						
VALEURSUIVANTE(x, S)						
VALEURPRÉCÉDENTE(x, S)						
MINIMUM(S)						
MAXIMUM(S)						

Remarques : RECHERCHER(x, S) renvoie la position de x dans S s'il est présent (et sinon NULL), les opérations SUPPRIMER(x, S), VALEURSUIVANTE(x, S), VALEURPRÉCÉDENTE(x, S) supposent que l'on connaît déjà la position de x dans S (appel à RECHERCHER(x, S) déjà fait).

Exercice 2.*Fusion de listes triées*

Pour des données munies d'un ordre total, décrire un algorithme en $O(n \log k)$ permettant de fusionner k listes triées en une seule liste triée, où n est le nombre total d'éléments de toutes les listes en entrée.

Exercice 3.*Range queries / requêtes sur intervalles*

Étant donné un tableau de n entiers, proposer un prétraitement dans chacun des cas suivants pour répondre aux requêtes avec les complexités annoncées.

1. Pour tout $1 \leq i \leq j \leq n$, être capable de calculer en temps constant le *minimum* des entiers du tableau compris entre les indices i et j , avec un prétraitement en temps et en espace $O(n^2)$.
2. Même question pour un calcul en temps $O(\log n)$, avec un prétraitement en temps et en espace $O(n)$.
3. Même question pour un calcul en temps $O(1)$, avec un prétraitement en temps et en espace $O(n \log n)$.
4. Pour tout $1 \leq i \leq j \leq n$, être capable de calculer en temps constant la *somme* des entiers du tableau compris entre les indices i et j , avec un prétraitement en temps et en espace $O(n)$.
5. Laquelle des solutions aux questions précédentes est la plus rapide à mettre à jour si des valeurs du tableau initial sont modifiées ?

Exercice 4.*Une alternative à l'initialisation*

Développer une technique qui permet de travailler sur un gigantesque tableau des entiers sans avoir à l'initialiser entièrement : les cellules du tableaux contiennent n'importe quoi au départ, on veut pouvoir traiter en $O(1)$ des opérations d'écriture ECRIRE(x, i) d'entier x à l'adresse i , et de lecture LIRE(i) qui renvoie l'entier stocké à l'adresse i si c'est bien le résultat d'une écriture précédente et sinon qui renvoie "cellule jamais initialisée".

Remarque : vous pouvez si vous le souhaitez introduire des structures de données annexes mais en ajoutant au plus un espace $O(\text{nb d'écritures})$.

Utilisation pratique : permet par exemple de travailler comme si tout le tableau avait été initialisé à 0 en interprétant une réponse "cellule jamais initialisée" comme 0.