

Série TD N° 5 : Les algorithmes de Tri et de recherche

On cherche à trier les valeurs d'une liste rangées dans une structure de tableau dans l'ordre croissant.

On pose pour la résolution des tous les exercices du cours et du TD la déclaration du tableau suivant :

TYEP TAB = tableau [1..100] d'entiers

Exercice 1 [Tri par dénombrement] :

Ce tri repose sur la connaissance préalable de la valeur maximale m contenue dans le tableau T . Le tri par dénombrement est réalisé en trois phases :

1. Initialisation : initialisation de toutes les cellules d'un tableau de distributions $D[1+m]$ à 0.
2. Dénombrement : parcours du tableau T et incrémentation de la cellule D dont l'indice est la valeur contenue dans la cellule courante du tableau T . A la fin, le tableau contient la distribution des valeurs de T
3. Tri : parcours du tableau D en rangeant simultanément la valeur de l'indice d'indexation i de D dans T autant de fois que $D[i]$.

Exemple : Soit $T = [3, 2, 1, 4, 4, 2, 1, 2, 6, 2]$. La valeur maximale dans T est $m = 6$. On initialise donc un tableau $D = [0, 0, 0, 0, 0, 0]$. Après la lecture des 9 valeurs du tableau T , on connaît la distribution des valeurs de T : $D = [2, 4, 1, 2, 0, 1]$. On a $D[5] = 0$ car la valeur T ne contient pas la valeur 5. On recopie donc deux valeurs de 1 dans T puis 4 valeurs de 2, etc. et T est alors trié.

Travail à faire :

1. Dérouler l'algorithme de tri par dénombrement sur le tableau :
 $A = [7, 1, 3, 1, 2, 4, 5, 2, 4, 3]$
2. Écrire une procédure pour calculer la fréquence des éléments du tableau A de n entiers positifs ayant des valeurs inférieures à k .
3. Écrire une procédure qui effectue le tri par dénombrement d'un tableau A de n entiers positifs ayant une valeur inférieure à k .

Exercice 2 [Tri par comptage]:

On se propose de trier, par ordre décroissant, un tableau T de taille N et dont ces éléments sont tous distincts, en utilisant la méthode de *tri par comptage*.

Le tri par comptage se déroule en deux étapes selon le principe suivant :

- 1^{ère} étape : *Création d'un tableau de compteurs*.

Chaque élément du tableau T à trier est comparé à tous les autres afin de déterminer le nombre d'éléments qui lui sont strictement supérieurs. Les résultats des N comptages sont rangés dans un tableau de compteurs, appelé C , possédant lui aussi N éléments. Les valeurs du tableau C sont comprises entre 0 et $(N - 1)$: le plus grand élément de T aura pour valeur du compteur 0 car il n'a aucun élément plus grand que lui et le plus petit élément aura pour valeur du compteur $(N - 1)$.

Exemple :

Tableau à trier :

$T :$	1	-9	0	5	13	22	2	10
-------	---	----	---	---	----	----	---	----

Tableau de compteurs :

$C :$	5	7	6	3	1	0	4	2
-------	---	---	---	---	---	---	---	---

- 2^{ème} étape : *Utilisation du tableau de compteurs pour le tri*.

Le premier élément du tableau T à trier, est permuté avec l'élément dont le compteur est égal à 0 . Les éléments correspondants dans le tableau de compteurs sont aussi permutés. Le deuxième élément du tableau T est permuté avec l'élément dont le compteur est égal à 1 et ainsi de suite. Ce processus est répété jusqu'à la valeur $(N - 1)$ de C .

Travail demandé :

1. Ecrire une procédure en Maple, appelée **COMPTAGE**, permettant de créer un tableau de compteurs C à partir d'un tableau T de taille N et dont les éléments sont tous distincts.
2. Ecrire une procédure en Maple, appelée **TRI_COMPTAGE**, permettant de trier par ordre décroissant un tableau T de taille N , selon le principe de tri détaillé précédemment, en utilisant la procédure **COMPTAGE** écrite en 1.

Exercice 3 [Fusionnement de deux tableaux triés]:

On dispose d'un tableau $T[g,d]$ et d'un indice p tel que les sous-tableaux gauche $T[g,p]$ et droit $T[p+1,d]$ sont tout deux triés. L'objectif de l'algorithme est de fusionner les deux sous-tableaux pour que la tableau $T[g,d]$ soit trié.

Description de l'algorithme à faire :

L'algorithme remplit un tableau auxiliaire S en prenant le contenu de la première cellule du tableau de gauche ou du tableau de droite si celle-ci est inférieure et ainsi de suite en avançant dans les sous-tableaux. Une fois les deux sous-tableaux fusionnés dans S , on recopiera les valeurs dans T . Cet algorithme est la clef du fonctionnement du Tri-Fusion. Il n'y a aucune difficulté mais il faut gérer correctement les indices qui vont balayer les sous-tableaux gauche et droit et le tableau auxiliaire S .

Exercice 4 : [recherches séquentiel les renvoyant un indice]

Nous avons vu comment implanter les algorithmes de recherche séquentielle pour réaliser des fonctions de recherche renvoyant une valeur booléenne. Vous allez dans cet exercice réaliser des fonctions donnant l'indice de l'élément trouvé.

1. Avant de vous lancer dans ces implantations, il faut préciser la spécification de ces fonctions de recherche.
 - a. Si l'élément recherché est présent plusieurs fois dans le tableau, quel indice renvoyer ?
 - b. Idem si cet élément n'est pas dans le tableau ?
2. Réalisez une implantation de la recherche séquentielle dans un tableau non trié, puis dans un tableau trié, qui donne le plus petit indice d'un élément présent dans le tableau.
3. Idem pour le plus grand indice

Exercice 5 : [Recherche dichotomique]

Pour comprendre la recherche dichotomique, soit $T = [0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22]$ un tableau d'entiers.

1. Donnez la suite des tranches de tableau parcourue lors de la recherche dichotomique de $x = 11$ dans le tableau T . Combien de comparaisons d'éléments du tableau sont-elles effectuées ?
2. Même question avec $x = 12$.
3. Même question avec $x = 10$.

4. Dessinez l'arbre des recherches possibles pour une recherche dichotomique dans un tableau de longueur 12.

Exercice 6 : [Recherche dichotomique renvoyant un indice]

1. Réalisez une implantation de la recherche dichotomique qui renvoie un indice plutôt qu'une valeur booléenne.
2. Dans le cas où l'élément recherché est présent plusieurs fois dans le tableau, quel est l'indice qui est renvoyé par votre fonction ?