Soit T[15] un tableau de 15 éléments tous entiers **distincts**

1. On suppose que T est trié dans le sens croissant à partir de la case 0 à la case m-1, m<15 . Soit J un entier,
   1. Fournir un algorithme qui permet d’insérer J dans T de telle sorte que T reste trié (jusqu’au [m]ème élément).
   2. Etablir un programme C qui réalise l’algorithme de a.(TP)
2. On suppose que T n’est pas trié
   1. Fournir un algorithme qui permet de trier le tableau T en utilisant la première question.
   2. Etablir un programme C qui réalise l’algorithme de a.
3. On suppose que T n’est pas trié:
   1. Fournir un algorithme qui permet de trier un tableau TAB en utilisant le principe de permutation
   2. Etablir un programme C qui réalise l’algorithme de a
4. Comparer les algorithmes de 3 et 4.
5. Fournir un algorithme qui permet de trier un tableau TAB en utilisant le principe de dichotomie
6. Soit T[5][5] un tableau à deux dimensions
   1. Fournir un algorithme qui permet d’intervertir les lignes et les colonnes de ce tableau puis le programme C
   2. Fournir un algorithme qui calcule le déterminant de ce tableau. puis le programme C
7. Soient Ta[5][5] et Tb[5][5] deux tableaux représentant deux matrices A et B
   1. Fournir la somme de ces deux matrices (algorithme et programme C)
   2. Fournir le produit de ces deux matrices (algorithme et programme
8. Fournir un algorithme qui effectue la recherche d’un élément dans Tab
   1. Dans le cas Tab non trié
   2. Dans le cas Tab trié
   3. Discuter de la complexité de chacun des deux cas précédents.
9. Fournir un algorithme qui effectue la suppression d’un élément dans Tab
   1. Dans le cas Tab non trié
   2. Dans le cas Tab trié
   3. Discuter de la complexité de chacun des deux cas précédents.
10. On se donne la matrice A

Et soient les représentations

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1-Ac | 1 | 6 | 3 | 4 | 6 | 7 | 5 | 9 |
| NumCl | 1 | 2 | 2 | 3 | 1 | 3 | 2 | 4 |
| NbNN | 2 | 2 | 2 | 2 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2-Ac | 1 | 6 | 3 | 4 | 6 | 7 | 5 | 9 |
| NumCl | 1 | 2 | 2 | 3 | 1 | 3 | 2 | 4 |
| NNNL | 2 | 4 | 6 | 8 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3-Ac | 1 | 6 | 3 | 4 | 6 | 7 | 5 | 9 |
| NumPl | 1 | 2 | 2 | 3 | 1 | 3 | 2 | 4 |
| NDebNL | 1 | 3 | 5 | 7 |

Montrer que les représentations A, 1-Ac , 2-Ac et 3-Ac sont équivalentes

Donner la représentation compacte de

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 3 | 0 | 4 | 0 | 0 | 0 |
| 0 | 0 | 0 | 4 | 9 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 2 | 0 | 8 | 0 |
| 4 | 0 | 4 | 0 | 7 | 0 | 0 | 5 |
| 1 | 0 | 3 | 0 | 0 | 6 | 0 | 4 |
| 0 | 0 | 5 | 0 | 6 | 0 | 0 | 8 |
| 0 | 5 | 0 | 5 | 0 | 0 | 7 | 0 |
| 4 | 0 | 0 | 5 | 0 | 0 | 8 | 0 |

Comparer les deux représentations, bidimensionnelle et compacte.

Fournir un algorithme qui permet de stocker compact une matrice donnée sous forme de tableau bidimensionnel et inversement.

Quels sont les cas où l’utilisation des stockages compacts est avantageuse ?Peut-on réaliser les autres opérations (addition de matrice, produit, multiplication par un vecteur, etc.) sur les matrices avec une représentation compacte ?