SESIÓN **/04**

ARREGLOS BIDIMENSIONALES

1. Características
2. Representación gráfica
3. Operaciones
4. Matrices especiales

**INTRODUCCIÓN**

* La clase anterior analizamos la eficiencia de los algoritmos de búsqueda y ordenamiento sobre los elementos de un Arreglo Unidimensional.
* En la presente sesión, veremos el segundo tipo de arreglo: los Arreglos Bidimensionales, conocidos también como Matrices. Revisaremos sus principales características, la forma cómo se representan y las operaciones que se pueden realizar sobre esta estructura.
* Culminaremos conociendo algunas matrices con características especiales: matriz cuadrada, matriz triangular superior y matriz triangular inferior.

1. **CARACTERÍSTICAS**

Las principales características de los Arreglos Bidimensionales ó arreglos de dos dimensiones, son las siguientes:

* Se les conoce también como Matrices.
* Se trata de una colección de valores del mismo **tipo de dato**, los cuales están distribuidos en filas y columnas.
* Para acceder a un elemento se necesita dos índices, el primero representa la fila y el segundo representa la columna.
* El primer elemento se almacena en la posición (0, 0).

1. **REPRESENTACIÓN GRÁFICA**

Un Arreglo Bidimensional o Matriz, se representa como una tabla ó cuadrilla, en la cual las filas y las columnas se numeran. Esta enumeración empieza en cero.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 |
| 0 |  |  |  |  |
| 1 |  |  |  |  |
| 2 |  |  |  |  |

1. **OPERACIONES**

Las operaciones que se pueden realizar sobre los datos almacenados en una Matriz son: recorrido, búsqueda, modificación, ordenamiento, inserción y eliminación.

**3.1 Recorrido**

Consiste en visitar cada elemento de la Matriz. El recorrido se puede realizar por filas ó por columnas. Si no se especifica, se asume la primera forma; es decir, recorrido por filas.

Primero se visita la posición (0, 0), luego la posición (0, 1) y así sucesivamente.

En el algoritmo para implementar esta operación, se utiliza estructuras repetitivas anidadas.

* 1. **Búqueda**

Se trata de encontrar un valor dentro de la Matriz. Se debe recorrer la Matriz comparando el dato buscado con cada elemento. El algoritmo termina cuando se encuentra el dato ó cuando se termina de recorrer la Matriz.

Si lo encuentra, el algoritmo devolverá la posición dónde está ubicado; es decir, el número de fila y el número de columna.

***EJEMPLO 1:***

***En la siguiente Matriz, busca el número 42***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 |
| 0 | 11 | 12 | 13 | 14 |
| 1 | 21 | 22 | 23 | 42 |
| 2 | 31 | 32 | 33 | 34 |

Primero se compara el número buscado con el elemento de la posición (0, 0). Como no son iguales, se compara con la posición (0, 1) y así sucesivamente hasta encontrar el número.

Haciendo el recorrido por filas, se tuvieron que realizar 8 comparaciones hasta encontrar el valor buscado.

***Resultado: El número 42 se encuentra en la posición (1, 3)***

* 1. **Modificación**

Consiste en reemplazar el valor de un elemento de la Matriz. Antes de realizar esta operación, primero se debe buscar el valor a reemplazar para obtener su posición.

***EJEMPLO 2:***

***En la siguiente Matriz, reemplaza el número 42 por el número 24***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 |
| 0 | 11 | 12 | 13 | 14 |
| 1 | 21 | 22 | 23 | 42 |
| 2 | 31 | 32 | 33 | 34 |

***Primero buscamos el número 42 dentro del Vector. Conociendo su posición, procedemos a realizar el reemplazo.***

* 1. **Ordenamiento**

Se trata de establecer un orden entre los valores de una fila ó de una columna. Este ordenamiento puede ser ascendente ó descendente.

Si no se especifica el tipo de ordenamiento, se asume ordenamiento ascendente (de menor a mayor).

* 1. **Inserción**

Consiste en agregar toda una fila ó toda una columna dentro de la Matriz. La inserción puede ser:

* Al inicio
* Al final
* Entre 2 filas ó entre 2 columnas

Cuando la inserción es por el final, se le denomina **Adicionar**.

* 1. **Eliminación**

Se trata de sacar toda una fila ó toda una columna de la Matriz. Al sacar elemento se libera el espacio de memoria que estaban ocupando. La eliminación puede ser:

* Al inicio
* Al final
* Entre 2 filas ó entre 2 columnas

1. **MATRICES ESPECIALES**

Son matrices que cumplen determinadas características, entre las más importantes tenemos las siguientes:

**4.1 Matriz cuadrada**

En este tipo de matrices, el número de filas es igual al número de columnas. Tienen una diagonal principal y una diagonal secundaria.

***EJEMPLO 1:***

***Se tiene la siguiente Matriz cuadrada de orden 3***

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

***Diagonal principal : conformada por los números 4, 5 y 6***

***Diagonal secundaria : conformada por los números 8, 5 y 2***

* 1. **Matrices poco densas**

Son matrices cuadradas que presentan una alta proporción de ceros entre sus elementos.

Las matrices poco densas más comunes son:

* Matriz triangular superior: Los elementos que se encuentran por debajo de la diagonal principal son ceros.

***EJEMPLO 2:***

***Se tiene la siguiente Matriz cuadrada de orden 4***

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

* Matriz triangular inferior: Los elementos que se encuentran por encima de la diagonal principal son ceros.

***EJEMPLO 3:***

***Se tiene la siguiente Matriz cuadrada de orden 4***

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

**CONCLUSIÓN**

Para acceder a cada elemento de una Matriz se utilizan dos índices: el primero para las filas y el segundo para las columnas. Las operaciones de ordenamiento, inserción y eliminación se realizan por filas ó por columnas. La eficiencia de los algoritmos aplicados a una Matriz, se mide de la misma forma que se mide en un Vector.

**REFERENCIAS BIBLIOGRÁFICAS**

* Cairo, O.; Guardati, S. (2008). Estructuras de datos. 3ra. Edición. México D.F., Mexico: McGraw Hill.
* Instituto NIIT (2011). Data Structures and Algorithms. Student guide.