SESIÓN / 02

OPERACIONES CON VECTORES

/ RECORRIDO
/ BÚSQUEDA
/ MODIFICACIÓN
/ ELIMINACIÓN

/ INTRODUCCIÓN

- La clase anterior conocimos las Estructuras de Datos, la forma cómo se clasifican y la importancia de una adecuada selección de los algoritmos para la elaboración de programas eficientes.

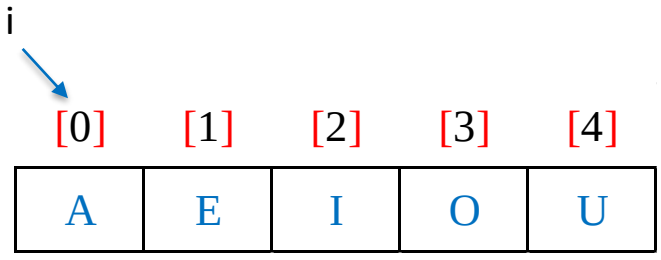
- En la presente sesión, trataremos con mayor detalle la primera Estructura de Datos: Los Arreglos; específicamente, los arreglos unidimensionales. Revisaremos sus principales características, la forma cómo se representan y las operaciones que se pueden realizar sobre esta estructura.

- Culminaremos analizando la eficiencia de los algoritmos de búsqueda y eliminación.

RECORRIDO

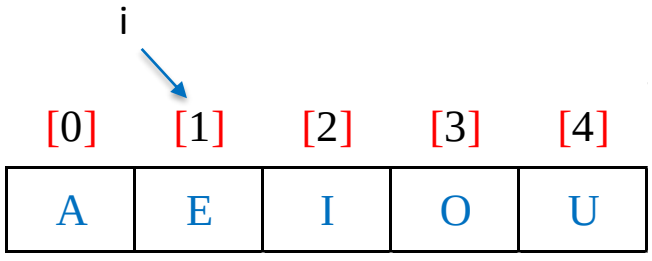
Recorrido

Consiste en visitar cada elemento del Vector.



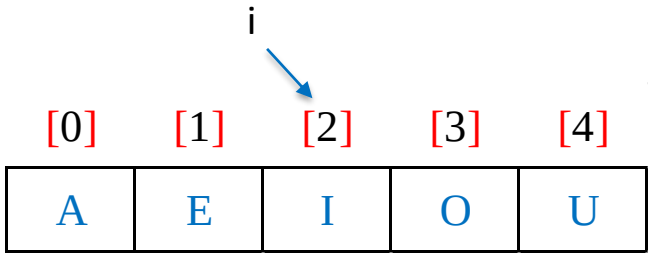
Recorrido

Consiste en visitar cada elemento del Vector.



Recorrido

Consiste en visitar cada elemento del Vector.



BÚSQUEDA

Búsqueda

Se trata de encontrar un valor dentro del Vector.

Ejemplo: En el siguiente Vector, busca el número 7

2	A[0]
9	A[1]
4	A[2]
7	A[3]
1	A[4]

Resultado:

El número 7 se encuentra en la posición 3

Algoritmos de búsqueda

- Búsqueda Lineal
- Búsqueda Binaria

/ BÚSQUEDA SECUENCIAL

Descripción

Consiste en recorrer el Vector, desde la posición cero, comparando cada elemento con el dato buscado.

Ejemplo: En el siguiente Vector, busca el número 70

A	0	1	2	3	4
	20	90	40	70	10

Resultado: El número 70 se encuentra en la posición 3. Se tuvieron que realizar 4 comparaciones.

Eficiencia

La eficiencia se determina por el número de comparaciones a realizar.

- Si el dato buscado se encuentra en la primera posición:

  Sólo se necesitará hacer una comparación.

Mejor caso

- Si el dato buscado se encuentra en la última posición ó

  si este no se encuentra en el Vector:

  Para un Vector con n elementos, se tendrá que realizar

  n comparaciones

Peor caso

/ BÚSQUEDA BINARIA

Descripción

Este algoritmo compara el dato buscado con el valor almacenado en la posición central del Vector. Si no son iguales, buscará el dato en la parte izquierda ó en la parte derecha del Vector, dependiendo del valor de la posición central.

Requisito: Los elementos del Vector, deben estar ordenados.

/ BÚSQUEDA BINARIA

Descripción

Ejemplo: Dado el siguiente Vector, busca el número 60

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
12	20	24	30	36	40	48	50	52	60	64	70	78	80	86	90

Nombre del Vector: Elemento

Dato a buscar: 60

Descripción

1. *Determinamos el límite inferior y el límite superior*

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
12	20	24	30	36	40	48	50	52	60	64	70	78	80	86	90

2. *Hallamos la posición central* $medio = \frac{0+15}{2}$ $medio = 7$

3. *Comparamos el dato buscado con el elemento de la posición central*

¿ dato = elemento[7] ?

¿ 60 = 50 ?

¿ dato < elemento[7] ?

¿ 60 < 50 ?

Descripción

1. *Determinamos el límite inferior y el límite superior*

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
12	20	24	30	36	40	48	50	52	60	64	70	78	80	86	90

2. *Hallamos la posición central* $medio = \frac{8+15}{2}$ *medio = 11*

3. *Comparamos el dato buscado con el elemento de la posición central*

¿ dato = elemento[11] ?

¿ 60 = 70 ?

¿ dato < elemento[11] ?

¿ 60 < 70 ?

Descripción

1. *Determinamos el límite inferior y el límite superior*

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
12	20	24	30	36	40	48	50	52	60	64	70	78	80	86	90

2. *Hallamos la posición central*

$$medio = \frac{8+10}{2} \qquad medio = 9$$

3. *Comparamos el dato buscado con el elemento de la posición central*

¿ dato = elemento[9] ?

¿ 60 = 60 ?

Eficiencia

La eficiencia se determina por el número de comparaciones a realizar.

- Si el dato buscado se encuentra en la posición central:

  Sólo se necesitará hacer una comparación.

Mejor caso

- Si el dato buscado se encuentra en uno de los extremos ó si este no se encuentra en el Vector:

  Para un Vector con n elementos, el número aproximado de comparaciones será: $\log_2 n$

Peor caso

MODIFICACIÓN

Modificación

Consiste en cambiar el valor de un elemento del Vector.

Ejemplo: En el siguiente Vector, modifica el número 9 por el número 8

2	A[0]
9	A[1]
4	A[2]
7	A[3]
1	A[4]

Modificación

Consiste en cambiar el valor de un elemento del Vector.

Ejemplo: En el siguiente Vector, modifica el número 9 por el número 8

2	A[0]
9	A[1]
4	A[2]
7	A[3]
1	A[4]

Resultado:

2	A[0]
8	A[1]
4	A[2]
7	A[3]
1	A[4]

ELIMINACIÓN

Eliminación

Se trata de sacar un elemento del Vector.

La eliminación puede ser:

- Al inicio.
- Al final.
- Entre dos elementos.

Eliminación

Se trata de sacar un elemento del Vector.

Ejemplo: En el siguiente Vector, elimina el número 9

2	A[0]
9	A[1]
4	A[2]
7	A[3]
1	A[4]

Eliminación

Se trata de sacar un elemento del Vector.

Ejemplo: En el siguiente Vector, elimina el número 9

2	A[0]
9	A[1]
4	A[2]
7	A[3]
1	A[4]

Resultado:

2	A[0]
4	A[1]
7	A[2]
1	A[3]
	A[4]

Se realizaron 3 traslados

/ CONCLUSIONES

- La forma de representar gráficamente un Arreglo Unidimensional depende de la operación que se desea realizar.
- Sobre los datos almacenados en un Vector, se pueden realizar seis operaciones.
- La eficiencia de los algoritmos de búsqueda se mide por el número de comparaciones a realizar.

/ BIBLIOGRAFÍA

- 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.