



FORMAS DE ALMACENAMIENTO DE ARREGLOS

PROF.: CRISPINA RAMOS S.

ARREGLOS EN MEMORIA

Hay varias formas de representar arreglos en memoria. Estas formas se pueden evaluar con base en cuatro características:

- a) Simplicidad de acceso a los elementos
- b) Facilidad de recorrer a través de varios caminos
- c) Facilidad de crecer
- d) Eficiencia de almacenamiento

ARREGLOS EN MEMORIA

LIMITE INFERIOR A UNO (1-N)

El almacenamiento para el elemento $V[l+1]$ será adyacente al almacenamiento del elemento $V[l]$ para $l = 1, 2, \dots, N-1$

Para obtener la dirección de inicio del elemento, es necesario conocer:

- Dirección de inicio del espacio reservado para almacenar el arreglo (B)
- Tamaño de cada elemento del arreglo (T)
- Cota o límite superior e inferior

GENERACIÓN DEL LIMITE INFERIOR ARREGLOS DE UNA DIMENSIÓN

Si B es la dirección de inicio del arreglo y T el tamaño de cada elemento e I es el elemento buscado, entonces la localidad del elemento buscado es:

$$B + (I - \text{Lim1}) * T$$

Ejemplo:

Si V[1:10], hallar la dirección del elemento V[6], y cuántos elementos lo preceden, si B=100 y T=2

GENERACIÓN DEL LIMITE INFERIOR ARREGLOS BIDIMENSIONALES

Estos pueden definirse de dos formas:

Un arreglo unidimensional en el que cada componente es otro tipo arreglo unidimensional

Un arreglo formado por un conjunto finito de elementos del mismo tipo y cada elemento está ordenado en dos dimensiones.

GENERACIÓN DEL LIMITE INFERIOR ARREGLOS BIDIMENSIONALES

El almacenamiento físico puede darse por fila o por columna, ya que se representa en forma lineal. Esto depende del lenguaje:

Por fila:

$$B + [(IF - CIF) * (CSC - CIC + 1) * T + (IC - CIC) * T]$$

Por columna:

$$B + [(IC - CIC) * (CSF - CIF + 1) * T + (IF - CIF) * T]$$

GENERACIÓN DEL LIMITE INFERIOR ARREGLOS BIDIMENSIONALES

Ejemplo:

Si tenemos la siguiente matriz:

$M[1:2][1:4]$

$B=200$; $T=2$

Encontrar: Dirección del elemento $M[2][4]$, cuántos elementos lo preceden? Por fila.

GENERACIÓN DEL LIMITE INFERIOR ARREGLOS MULTIDIMENSIONALES

Para arreglos de 3 dimensiones

Por fila:

$$B + T * [((E1L2 + E2)L3 + E3)]$$

Por columna:

$$B + T * [((E3L2 + E2)L1 + E1)]$$

Para más de tres dimensiones la fórmula se va a extender dependiendo del almacenamiento.

NOMENCLATURA

E= Posición ; L= longitud

E1=Fila , E2=Columna, E3= Profundidad

L1=Fila , L2= Columna, L3= Profundidad

E1=IF-CIF

L1=CSF-CIF+1

E2=IC-CIC

L2=CSC – CIC +1

E3=IP-CIP

L3=CSP-CIP+1