



Ciclo 1

Semana 6

Estructuras de Datos Lineales

Lectura 3 – Almacenamiento de Arreglos en Memoria



| Almacenamiento de Arreglos en Memoria

Debido a la importancia de los arreglos, casi todos los lenguajes de programación de alto nivel proporcionan medios eficaces para almacenar y acceder a los elementos de los arreglos, de tal forma que el programador no deba preocuparse de los detalles específicos de su almacenamiento, el cual está dispuesto fundamentalmente en secuencia contigua, por ejemplo, para un vector sería así (ver figura 1):

Si cada elemento del vector ocupa S bytes (1 byte 8 bits) y B es la dirección inicial como se muestra en la figura 1, la dirección inicial del elemento i -ésimo sería: $B + (i - 1) * S$.

En el caso de arreglos bidimensionales, los lenguajes de programación los almacenan en memoria de dos formas: orden de fila mayor y orden de columna mayor (figura 3), siendo el primero el medio más natural en que se leen y almacenan estos arreglos. Ver figura 2.

Posición B	$A[1]$
Posición $B + 1$	$A[2]$
\vdots	
	$A[3]$
	\vdots
	$A[i]$
	\vdots
Posición $B+n-1$	$A[n]$

Figura 1: tomado de Joyanes



Semana 6

Estructuras de datos lineales



Figura 2: tomado de Joyanes

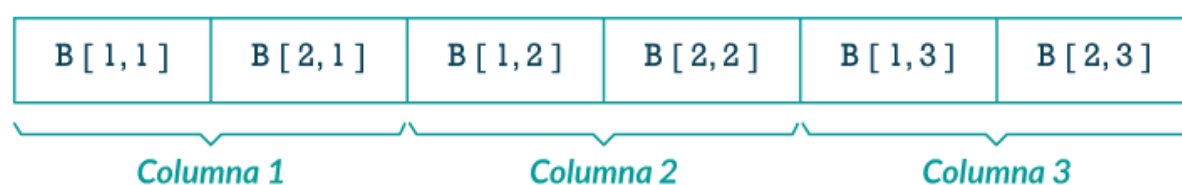
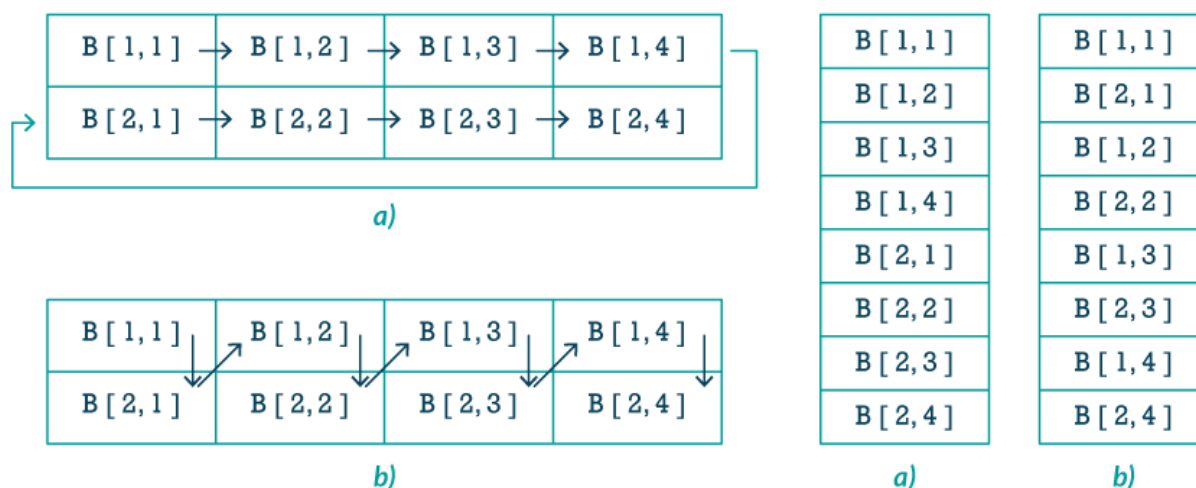


Figura 3: tomado de Joyanes

De modo general el compilador del lenguaje de alto nivel debe ser capaz de calcular con el índice $[i][j]$ la posición del elemento correspondiente. En el orden de fila mayor, cuyos subíndices máximos sean m y n (m filas, n columnas), la posición p del elemento $[i][j]$ con relación al primer elemento es: $p = n(i - 1) + j$.

Para calcular la dirección real del elemento $[i][j]$ se añade p a la posición del primer elemento y se resta 1, ver figura 4.



Almacenamiento de una matriz: a) por filas, b) por columnas

Figura 4: tomado de Joyanes