Data Structures (DS)
GTU # 3130702

Unit-2

Linear Data Structure

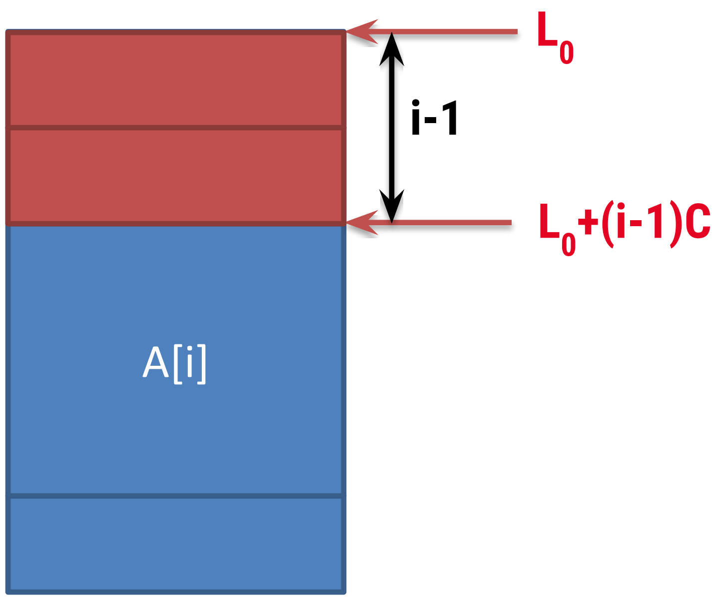
Computer Engineering Department SVIT Vasad

Outline

- Array
  - Representation of arrays
    - One dimensional array
    - Two dimensional array
- Applications of arrays
  - Symbol Manipulation (matrix representation of polynomial equation)
  - Sparse matrix
- Sparse matrix and its representation

One Dimensional Array

- Simplest data structure that makes use of computed address to locate its elements is the one-dimensional array or vector.
- Number of memory locations is sequentially allocated to the vector.
- A vector size is fixed and therefore requires a fixed number of memory locations.
- Vector A with subscript lower bound of “one” is represented as below….



- L_0 is the address of the first word allocated to the first element of vector A
- C words is size of each element or node
- The address of element Ai is

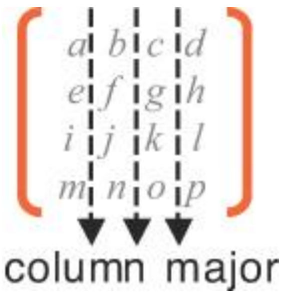
$$Loc(Ai) = L_0 + (C*(i-1))$$

- Let’s consider the more general case of a vector A with lower bound for it’s subscript is given by some variable b.
- The address of element Ai is

$$Loc(Ai) = L_0 + (C*(i-b))$$

Two Dimensional Array

- Two dimensional arrays are also called ***table*** or ***matrix***
- Two dimensional arrays have two subscripts
- **Column major order matrix:** Two dimensional array in which elements are stored column by column is called as column major matrix
- Two dimensional array consisting of **two rows** and **four columns** is stored sequentially by columns : A[1,1], A[2,1], A[1,2], A[2,2], A[1,3], A[2,3], A[1,4], A[2,4]



	Col-1	Col-2	Col-3	Col-4
Row 1	[1,1]	[1,2]	[1,3]	[1,4]
Row 2	[2,1]	[2,2]	[2,3]	[2,4]

Column major order matrix

$$
\begin{array}{cc}
 & \begin{array}{cccc} \text{Col-1} & \text{Col-2} & \text{Col-3} & \text{Col-4} \end{array} \\
\begin{array}{c} \text{Row 1} \\ \text{Row 2} \end{array} & \begin{bmatrix} [1,1] & [1,2] & [1,3] & [1,4] \\ [2,1] & [2,2] & [2,3] & [2,4] \end{bmatrix}
\end{array}
$$

- The address of element A [i , j] can be obtained by expression

  $Loc(A[i,j]) = L_0 + (j-1)*2 + (i-1)$

  $Loc(A[2,3]) = L_0 + (3-1)*2 + (2-1) = \mathbf{L_0 + 5}$

- In general for two dimensional array consisting of **n rows** and **m columns** the address element A [i , j] is given by

  $Loc(A[i,j]) = L_0 + (j-1)*n + (i-1)$

Row major order matrix

- **Row major order matrix:** Two dimensional array in which elements are stored row by row is called as row major matrix

$$
\begin{array}{c|cccc}
 & b2 & & & u2 \\
\hline
b1 & [1,1] & [1,2] & [1,3] & [1,m] \\
 & [2,1] & [2,2] & [2,3] & [2,m] \\
 & & & & \\
u1 & [n,1] & [n,2] & [n,3] & [n,m]
\end{array}_{\text{nxm}}
$$

n = no of rows, **m** = no of columns

b1 = lower bound subscript of row
u1 = upper bound subscript of row
n = u1 – b1 + 1

b2 = lower bound subscript of column
u2 = upper bound subscript of column
m = u2 – b2 + 1

- The address element A [i , j] is given by

$$\text{Loc}(A[i,j]) = L_0 + (i-1)*m + (j-1)$$

- The address element A [i , j] is given by

$$\text{Loc}(A[i,j]) = L_0 + (i-b1)*(u2-b2+1) + (j-b2)$$

Applications of Array

1. Symbol Manipulation (matrix representation of polynomial equation)
2. Sparse Matrix

- Matrix representation of polynomial equation
  - We can use array for different kind of operations in polynomial equation such as addition, subtraction, division, differentiation etc…
  - We are interested in finding suitable representation for polynomial so that different operations like addition, subtraction etc… can be performed in efficient manner.
  - Array can be used to represent Polynomial equation.

Address Calculation Problems

- A matrix **P[15][10]** is stored with each element requiring **8 bytes** of storage. If the base address at **P[0][0] is 1400**, determine the address at **P[10][7]** when the matrix is stored in Row Major Wise.

- **Solution:**
  - The given values are: Lo = 1400, b1 = 0, u1 = 14, b2 = 0, u2 = 9, c = 8, i = 10, j=7
  - **Row Major**
  - **Address of Loc(Pij) = Lo + [C * ((i – b1) (u2 – b2 + 1) + (j – b2))]**
  - **Loc($P_{10,7}$) = 1400 + [8 * ((10 – 0) (9 – 0 + 1) + (7 – 0))]**
  - **= 1400 + [8 * ((10) (10) + (7))]**
  - **= 1400 + [8 * (100 + 7)] = 1400 + [8 * 107]**
  - **= 1400 + [856] = 2256**

Address Calculation Problems

- The array **A[-2…10][3…8]** contains double type elements require **8 bytes** to store. If the **base address** is **4110**, find the address of **A[4][5],** when the array is stored in **Column Major Wise**.
- **Solution:**
  - The given values are: Lo = 4110,   b1 = -2, u1 = 10,   b2 = 3, u2 = 8,   c = 8,   i = 4, j=5
  - **Row Major**
  - **Address of Loc(Aij) = Lo + [C * ((j – b2) (u1 – b1 + 1) + (i – b1))]**
  - **Loc($A_{4,5}$) = 4110 + [8 * ((5 – 3) (10 – (-2) + 1) + (4 – (-2)))]**
  - **= 4110 + [8 * ((2) (13) + (6))]**
  - **= 4110 + [8 * (26+6)]   = 4110 + [8 * 32]**
  - **= 4110 + [256] = 4366**

Representation of Polynomial equation

	Y	Y^2	Y^3	Y^4
X	XY	XY^2	XY^3	XY^4
X^2	X^3Y	X^2Y^2	X^2Y^3	X^2Y^4
X^3	X^3Y	X^3Y^2	X^3Y^3	X^3Y^4
X^4	X^4Y	X^4Y^2	X^4Y^3	X^4Y^4

$2X^2 + 5XY + Y^2$

		Y	Y^2	Y^3	Y^4
	0	0	1	0	0
X	0	5	0	0	0
X^2	2	0	0	0	0
X^3	0	0	0	0	0
X^4	0	0	0	0	0

$X^2 + 3XY + Y^2 + Y - X$

		Y	Y^2	Y^3	Y^4
	0	1	1	0	0
X	-1	3	0	0	0
X^2	1	0	0	0	0
X^3	0	0	0	0	0
X^4	0	0	0	0	0

Sparse matrix

- An m x n matrix is said to be ***sparse*** if “many” of its elements are zero.
- A matrix that is not sparse is called a ***dense matrix***.
- We can device a simple representation scheme whose space requirement equals the size of the non-zero elements.

$$
\begin{array}{r|cccccccc|}
 & \text{Column - 1} & \text{Column - 2} & \text{Column - 3} & \text{Column - 4} & \text{Column - 5} & \text{Column - 6} & \text{Column - 7} & \text{Column - 8} \\
\hline
\text{Row - 1} & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 \\
\text{Row - 2} & 0 & 6 & 0 & 0 & 7 & 0 & 0 & 3 \\
\text{Row - 3} & 0 & 0 & 0 & 9 & 0 & 8 & 0 & 0 \\
\text{Row - 4} & 0 & 4 & 5 & 0 & 0 & 0 & 0 & 0 \\
\hline
\end{array}_{4x8}
$$

Terms	0	1	2	3	4	5	6	7	8
Row	1	1	2	2	2	3	3	4	4
Column	4	7	2	5	8	4	6	2	3
Value	2	1	6	7	3	9	8	4	5

Linear Representation of given matrix

Sparse matrix Cont…

- To construct matrix structure from liner representation we need to record.
  - Original row and columns of each non zero entries.
  - Number of rows and columns in the matrix.
- So each element of the array into which the sparse matrix is mapped need to have three fields: ***row***, ***column*** *and* ***value***

Sparse matrix Cont...

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array}\begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 0 & 0 & 6 & 0 & 9 & 0 & 0 \\ 2 & 0 & 0 & 7 & 8 & 0 & 4 \\ 10 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 12 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 5 \end{pmatrix}_{6x7} \end{array}$$

Memory Space required to store 6x7 matrix

42 x 2 = 84 bytes

Memory Space required to store Linear Representation

30 x 2 = 60 bytes

Linear representation of Matrix

Row	Column	A
1	3	6
1	5	9
2	1	2
2	4	7
2	5	8
2	7	4
3	1	10
4	3	12
6	4	3
6	7	5

Space Saved = 84 – 60 = 24 bytes

Sparse matrix Cont…

Linear Representation of Matrix

Row	Column	A
1	3	6
1	5	9
2	1	2
2	4	7
2	5	8
2	7	4
3	1	10
4	3	12
6	4	3
6	7	5

Linear Representation of Matrix

	Row
1	1
2	3
3	7
4	8
5	0
6	9

	Column	A
1	3	6
2	5	9
3	1	2
4	4	7
5	5	8
6	7	4
7	1	10
8	3	12
9	4	3
10	7	5

Memory Space required to store Liner Representation = 26 x 2 = 42 bytes

Thank You