Bharathidasan University

Centre for Differently Abled Persons

Khajamalai Campus

Tiruchirappalli-620 023

Tamilnadu

Bachelor of Computer Applications

(For Students with Speech and Hearing Impairment)

Course: Fundamentals of Data Structure

unit-1

Compiled By

Dr.M.Prabavathy

(Assistant Professor

Ms V.Vijayalakshm

Fundamentals of Data Structure

UNIT 1:

Definition of Data Structure

- The Logical (or) Mathematical Model of a particular organization of data is called data structure.

Primitive and Composite Data Types Primitive Data Types

- Number
- String
- Boolean
- Character —char( )
- Integer — int( ) , short( ) , long( ) ,byte( )
- Float — float( ) , double( )

Composite Data Types

- Arrays
- Records
- Matrices

Asymptotic Notation:

1) Logarithmic Function — Log n an +b
2) Linear Function — an^A2 +bn +c
3) Quadratic F unction — an^A2+....+ an^Az + $a*n^A1$
4) Polynomial Function Where z is some constant. — +$a*n^A0$ aAn
5) Exponential Function

where a is some constant.

Arrays

Array Definition:

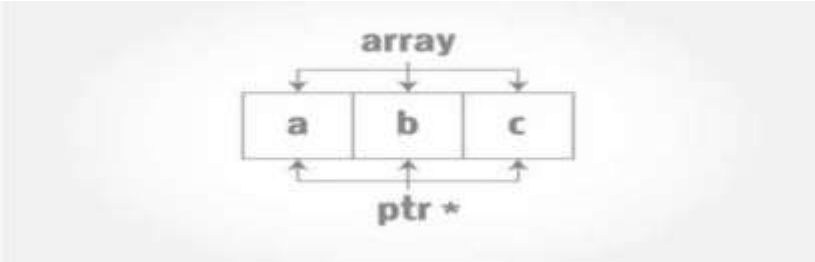
- Arrays means sequential memory location these linear structure are called Arrays.

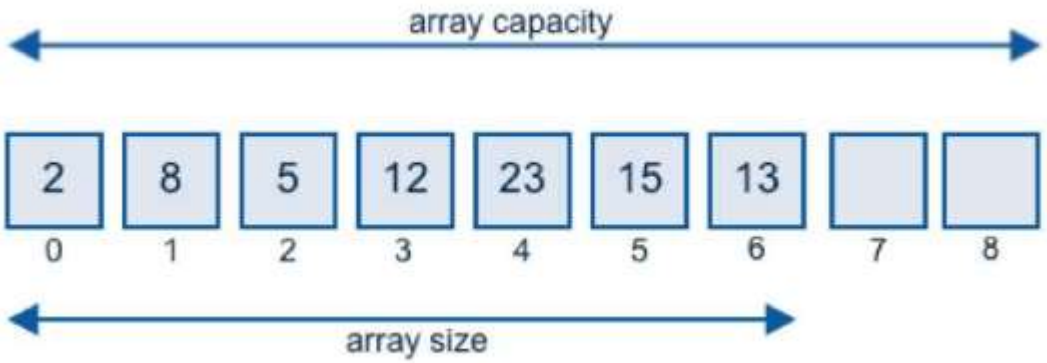
  Eg: A[1], A[2], A[3].....A[n].

elements	35	33	42	10	14	19	27	44	26	31
index	0	1	2	3	4	5	6	7	8	9

i-1

Size :10





Types of Arrays:

1. One Dimensional Array

5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C

A one memory location of elements is stored is called one dimensional array.

Eg: A[1], A[2],A[3]......A[N].

One Dimensional Array

Example:

Grace [O] = 95 Grade [1] = 85 Grace [2] =75 Grace **f** 3] =1 OO Grace [-4] = 65

2. Two Dimensional Array

5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C5*C

A two dimensional m*n array A is a collection of m.n(TABLE FORMAT) rows and columns data elements such that pair of subscripts integers.

Eg: A[1][1], A[1]A[2], A[1]A[3]   A[1]A[N]

Two-Dimensional Arrays

- A *one-dimensional array* stores a list of elements
- A *two-dimensional array* can be thought of as a table of elements, with rows and columns

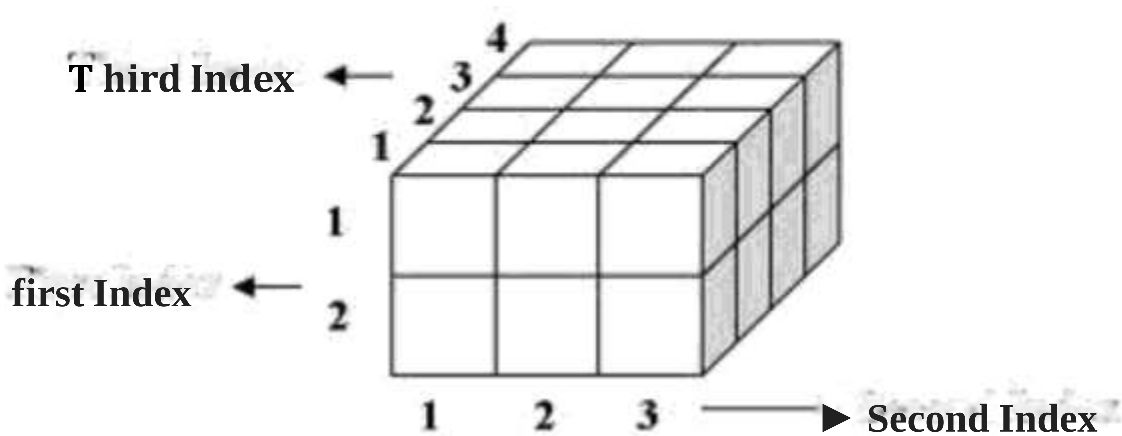
one dimension

two dimensions

	Column 0	Column 1	Column 2	Column 3
Row 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Row 1	a [1] [0]	a[1][1]	a[1][2]	a[1] 3]
Row 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]

3.Three Dimensional Array (or) Multi Dimensional Array

A is a collection of more than data elements such that pair of Subscripts integer.



Three-dimensional array with twenty four elements

Eg:	(1,1)	(1,2)	(1,3)-	••.(1,n)
	(2,1)	(2,2)	(2,3) ...	..(2,n)
	(3,1)	(3,2)	(3,3)...	....(3,n)

Linear Array:

- A linear Array is a list of a finite number (n) of homogeneous data elements.
- Length=UB-LB+1.
- UB -Upper Bound
- LB —Lower Bound

Linear	Non-Linear
Pointers	Trees
Linked List	Graphs

Linear Arrays

Example :

A linear array DATA consisting of the name of six elements

DATA

1	247
2	56
3	429
4	135
5	87
6	156

DATA[1] = 247

DATA[2] = 56

DATA[3] = 429

DATA[4] = 135

DATA[5] = 87

DATA[6] = 156

Data Structure Operations

1. Traversing (Process)
2. Searching (find)
3. Inserting (Add)
4. Deleting (Remove)
5. Sorting (Arrange)
6. Merging (Combine)

Operations on Arrays

❖ Create : create a new array elements

Create(LA,LB,UB)

STEP 1: Let LA[10],LB,UB,K STEP 2: LB=1 UB=10 STEP 3: Display Enter 10 Elements STEP 4: Repeat for K=LB to UB Increment by 1 Input LA[K]

[End of for]

STEP 5: Exit.

Eg:

1	2	3	4	5
10	20	30	40	50

❖ Traversing : Print all the Array Elements one by one

Traversal(LA,LB,UB)

STEP 1: Repeat for K=LB to UB by 1 Apply process to

LA[K]

[End of for]

STEP 2: Exit.

Eg:

1	2	3	4	5
10	20	30	40	50

Searching : Search an Element by the Value

Search (LA,LB,UB,ITEM)

STEP1 : Let K , set K=LB STEP 2: Repeat

while (K<=UB) LA[K]=ITEM, then

Display “ITEM found at K”

Set K=K+1

STEP 3: Exit.

Eg:

1	2	3	4	5
10	20	30	40	50

Inserting : Add an Element at the given index

Insert (LA,LB,UB,ITEM,POS)

STEP 1: Let K

STEP 2: Repeat for K=UB to POS Decreasing

by 1 LA[K+1]=LA[K]

LA[P0S]=ITEM

Set UB=UB+1

STEP 3: Exit.

Eg:

1	2	3	4	5
10	20	30	40	50

6	7
60	70

❖ Deleting : Delete an Element at the given index

Deleters (LA,LB,UB,POS)

STEP 1: Let K

STEP 2 : Repeat for K=POS to UB increment by 1

LA[K]=LA[K+1]

UB=UB-1

STEP 3: Exit.

Eg:

1	2	3	4	5
10	20	30	40	50

Eg:

1		3	4	5
10		30	40	50

❖ Sorting :Sort on Element at the given index

Eg:

1	2	3	4	5
50	30	10	20	40

Eg:

1	2	3	4	5
10	20	30	40	50

❖ **Merging : Combine an Element at the given index**

Eg:

1	2	3	4	5
10	70	30	40	60

6	7
50	20

Eg:

1	2	3	4	5	6	7
10	20	30	40	50	60	70

Thank You