SEPTEMBER 2017

M T W T F S S M T W T F S S

1 2 3 4 5 6 7 8 9 10

11 12 13 14 15 16 17 18 19 20 21 22 23 24

25 26 27 28 29 30

SATURDAY 12 AUGUST

WK 32 • 224-141

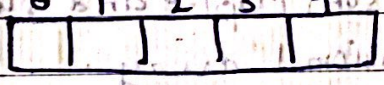
Unit-II

Arrays, Stacks & Queues

Arrays

- Arrays are the collection of a finite number of homogeneous data elements.

- Elements of the array are referenced respectively by an index set consisting of n consecutive numbers and are stored respectively in successive memory locations.



- The number n of elements is called the length or size of the array.

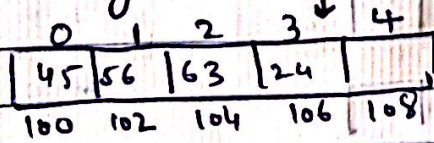
Length = UB − LB + 1

where UB is the largest index, called upper bound

LB " " smallest " , called lower bound

SUNDAY 13

- To access an array element, address of an element is computed as an offset from the base address of the array and one multiplication is needed to compute what is suppose to be added to the base address to get the memory address of the element.



a[3] = 20 17

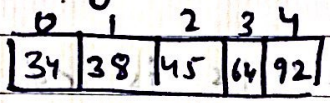
Base add. = 100 + Size of (int) * 3

= 100 + 6 = 106

→ This process takes one multiplication and one addition. Since these two operation take constant time, we can say the array access can be performed in constant time.

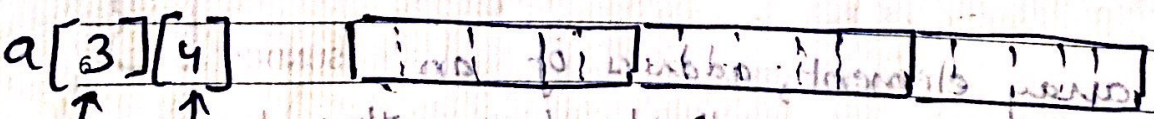
Array Dimensions

1d array — A list of data items that can be represented by one variable name using only one subscript and such variable is called one dimensional array.

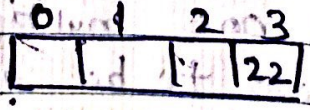


a[0] = 34

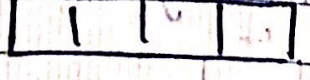
2d array — A list of data items that can be represented as rows and columns, which by one name and two subscripts

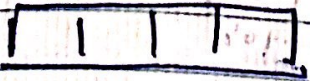


or



eg: b[0][3] = 22





Basic operations in Linear Arrays: Traversing, Inserting, Deleting

(1) Traversing Linear Array (To access and process each element of array exactly once.)

- If array is - LA
- Upper Bound - UB
- Lower Bound - LB
- Applying an operation - PROCESS

Algorithm:
(1) Initialize counter K := LB
(2) Repeat steps 3 and 4 while K ≤ UB
(3) Apply PROCESS to LA[K]
(4) Increase counter K = K+1
(5) End loop
(6) Exit

eg. We have array

0	1	2	3	4
2	3	5	8	6

LB ↑ (0) UB ↑ (4)

Process is to multiply each element by 2.

(1) K = 0
(2) while K ≤ 4
(3) Process LA[0] = 2 × 2 = 4
(4) K = K+1 = 0+1 = 1
(5) End loop
(6) Exit

WEDNESDAY
16
AUGUST
228-137 • WK 33

JULY 2017

S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

(2) Inseting (Refers to the operation of adding another element to the collection.)

→ Inseting an element in a linear Array of N elements at Kth location.

Algorithm:

(1) Initialize a counter I = N

(2) loop (while I ≥ K)

(3) LA[I+1] = LA[I]

LA[5] = LA[4]

(4) I = I-1 LA[4] = LA[3]

I = 3

(5.) End loop

(6) Set LA[K] = item

LA[3] = E

(7) N = N+1

Array size will be 5

(8) Exit

1	A
2	B
3	C
4	D
5	
6	

I = 4

E

I have to Insert E at location 3.

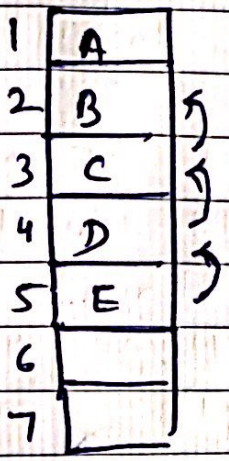
Now array will become

1	A
2	B
3	E
4	C
5	D

20
17

(3) Deleting (Refers to the operation of removing one of the elements)

eg.



. I have to delete B. Then C has to shift up & D and E too.

Algorithm: (1) Set Item = LA[K]
= LA[2]

(2) Loop for I = K to N-1
I = 2 to (5-1) = 2 to 4

(3) Set LA[I] = LA[I+1]
LA[2] = LA[3]

(4) End loop LA[3] = LA[4]
LA[4] = LA[5]

(5) N = N-1
= 4

(6) Exit