DATE : __/__/__
PAGE : _____

15 apr 2023 Data Structure for Gate

by Vishvadeep gothi sir

Mathematical and logical model of organizing the interrelated data.

- organizing → predefined structure to organize the data
  for ex: Queue - FIFO
  Stack - LIFO
- interrelated → operations on data

It is a way to organize the data on the computer so that it can be accessed and update efficiently.

Types of data structure

* Linear - Elements are arranged in sequential fashion.
  ex Array, linked-list, Queue, Stack
* non linear - Elements are arranged in non linear fashion
  ex: Tree, graph, Hash table

Classification of data structure

- Container
  ex Array
  Linked list
- Priority
  ex: queue
  Stack
  heap
- Indexing
  BST
  B-tree
  B+
  Hashtable

DATE : / /
PAGE :

Syllabus

1) Array
2) Linked list
3) Queue
4) Stack
5) Expression
   Recursion
6) Tree
7) Graph
8) Hashing

Operation on data structure

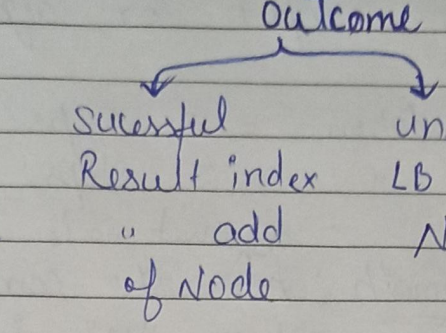
1) Traversing: Visiting each and every element at-least & atmost once.

2) Insertion: Adding an element in a data structure
   Overflow error: The data structure is already full and we are trying to insert more element

3) Deletion: Removing an element from in ds.
   underflow: The data structure is empty and we are trying to delete an element.

4) Searching: Finding an element in a ds.

output : Location of that element



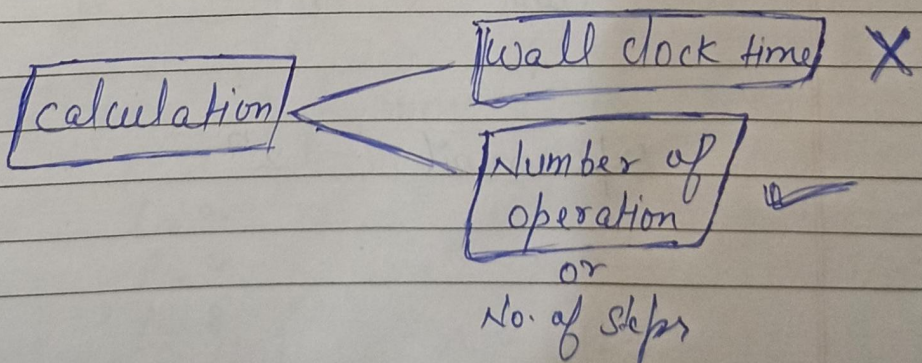
5) Merging : Combining 2 data structure of same type into one

6) Sorting : Arranging the elements of asc or dsc manner

Lecture2 - Algorithm & Analysis of Algorithm

Analysis : when we need to compare multiple solution for a problem.

1) Space complexity : Space needed to run algo (expect input size)
2) Run-Time complexity : Time needed to run algo.



DATE : __/__/__
PAGE :

Assume : $n=2$ input for algo | $n=10$

Algo 1 taken = 2 steps | 10 steps
" 2 " = 3 steps | 3 steps
at what rate

Rate of growth : how the complexity increase when "n" increases.

$T(n) = ?$

n	T(n)
1	2
2	3
3	4
4	5
⋮	⋮
100	100

$T(n) = n+1$

Complexity Notation :

Constant - (1) Complexity is not dependent on n
logarithmic - $\log n$
linear - n
log linear - $n \log n$
quadratic - n^2
cubic - n^3
⋮
exponential - 2^n

DATE : __/__/__
PAGE : _____

ex:

① `for(i=1; i<=n; i++){` — $n+1$

`x = m+2` — n

`}`

$2n+1$

Complexity = n (linear)

② `for(i=1; i<=n; i++){` — $n+1$ = $n+1$

`for(j=1; j<=n; j++){` — $n(n+1) = n^2+n$

`x = m+2` — $n*n = n^2$

`}`

`}`

$2n^2+2n+1$

Complexity = $\theta(n^2)$

Ex 3

```
for(i=1; i<n; i=i*2){
    x = m+2
}
```

$\log_2 n$

```
for(i=n; i>1; i=i/2){
    x = m+2
}
```

$\log_2 n$

Let's assume, loop runs x times

$i=1$
$i=2$
$i=4$
$i=8$
$i=16$

$n=32$

$2^x < n$

$x = \log_2 n$

DATE : / /
PAGE :

Asymtotic Notation

1) Big Oh - It gives tightest upper bound.
2) Omega Ω - It gives tightest lower bound.
3) Theta Θ - Avg bound (exact bound)

Best & worst case - (Type of input)

Lecture 3 - Array Introduction

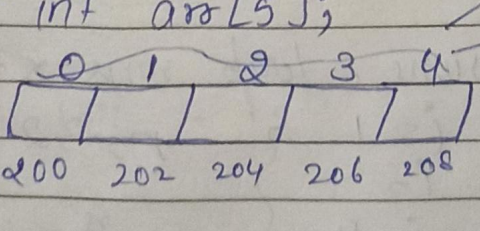
* Array : is a collection homogeneous elements (similar type)

* Characteristics of Array
  - All the elements stored on consecutive memory location
  - All the elements can be accessed using a set of index

* Defining Array : in c language

datatype name[size];

ex : int arr[5];



Lower bound (LB) = 0
upper " (UB) = 4

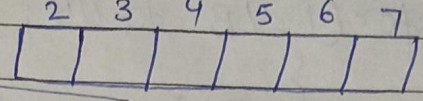
DATE : __/__/__
PAGE : _____

* Defining Array : In data structure

name[LB : UB] $LB \leq UB$

ex: arr[0:4] size 5

B[2:7] ↔ array B

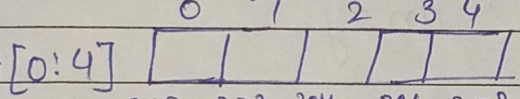


C[-6 : 8] 8-(-6)+1 = 15

Size of array = UB - LB + 1

array[LB : UB] ⇒ array[LB ... UB]

* Location of an array element



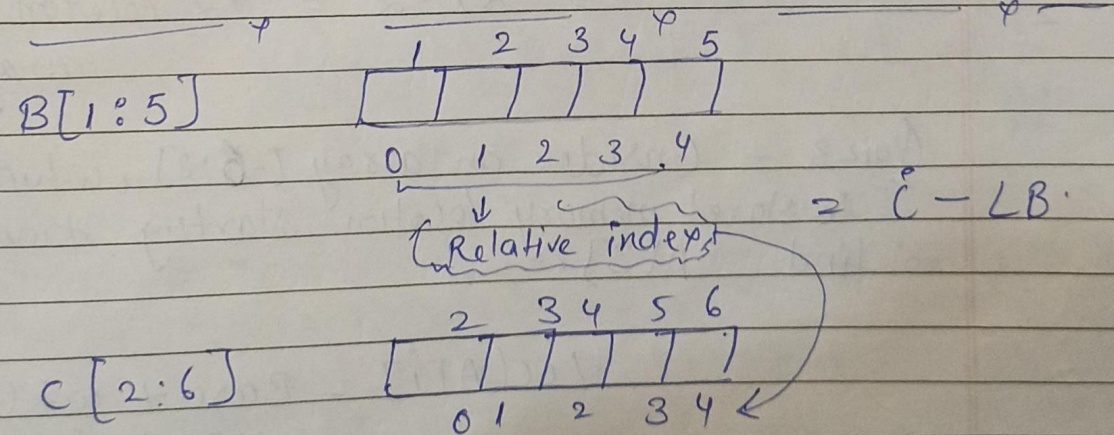
Loc[A[i]] = Base + w * Relative index of element with index i

Assume each element occupies 2 locations in memory (w = 2)

find relative index i.no. of elements in array before the element

Eg:

Loc[A[3]] = 200 + 2 * 3 = 206



$Loc[A[i]] = w(i - LB) + Base$

* Traversing In Array

```
for{i = LB to UB, i++
    {
        Process arr[i]
    }
```

if array has n elements = $O(n)$

UB - LB + 1
18 - 6 + 1
(13)

Quiz 1

Consider an array A[6:18], total no. of elements in array

$UB - LB + 1 \Rightarrow 18 - 12 + 1 = 13$

② Consider an array A[-8:12], each takes 2 locations in memory, Total space needed to store array

$Size = 12 - (-8) + 1 = 21$

$= 21 * 2 = 42$ location in array

Quiz 3 - Consider an array [-6:8], which is stored memory location starting from 1000 find A[3]?

$Loc(A[i] = Base + w*(i - LB)$

$1000 + 4(3 - (-6)$

$= 1036$ Ans

DATE : __/__/__
PAGE : _____

Lecture 4 : Maximum and minimum of array

Finding Minimum

(1) 15, 8, 9, 9, 7, 12, 19, 20, 4, 21, 2, 11, 6, 1, 5

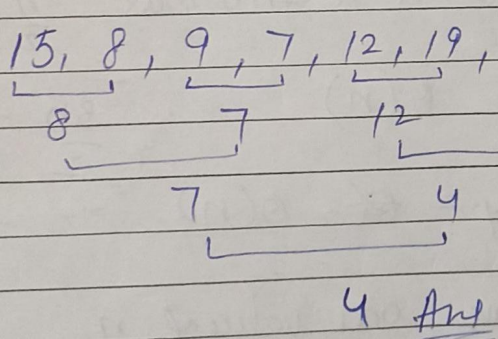
min = ~~15~~ ~~8~~ ~~7~~ ~~4~~ ~~2~~ 1

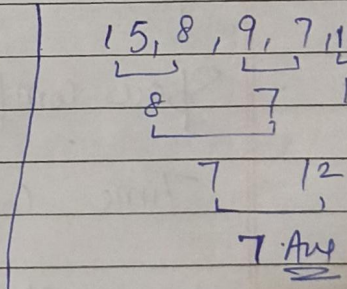
no. of comparisions = n-1

```
min = A[LB]
for(i=LB+1; i<=UB; i++) {
    if(A[i] < min)
        min = A[i]
}
return min
```

Time complexity = $\Theta(n)$
Space " = $\Theta(1)$ or $O(1)$

(2) Tournament method of finding minimum





no. of comparisions = n-1, Time com = $\Theta(n)$
Space complexity = $\Theta(n)$

DATE : / /
PAGE :

* Finding Maximum

15, 8, 9, 7, 12, 19, 20, 4, 21, 2, 11, 6, 1, 5.

max : ~~15~~ ~~19~~, ~~20~~, 21

Time Complexity = $O(n)$
no. of comparison = $n-1$
Space = $O(n)$

Using Tournament

no. of Comp = $n-1$
Time Complexity = $O(n)$
Space " = $O(n)$

Find Minimum & Maximum with minimum Comparisons

5, 3, 1, 6, 9, 4

3	$\frac{n}{2}$	5	$\frac{n}{2}$
1		6	
4		9	

findMin findMax

To make 2 list no. of Comparisons are $\frac{n}{2}$

To find min $\frac{n}{2} - 1$

To find max $\frac{n}{2} - 1$

$\frac{3n}{2} - 2$

Space complexity = $O(n)$

Time Complexity = ~~$O(3$~~ $O(n)$

Note : If we have add value of n,
no. of Comparisions = $3\left[\frac{n}{2}\right] - 2$

---

first find min = $n-1$
then find max = $n-1$

$2n-1$

DATE : __/__/__
PAGE : _____

* Find Second Minimum

Comparisions	Space
first find min = $n-1$	O(1)
then find min of remaining = $n-2$	
$2n-3$	

* Find Second Minimum with minimum comparisons

5, 9, 4, 2, 6, 1, 3, 7

5 2 1 3

2 1

1

Ans

but this method will no work always

5, 9, 4, 3, 6, 1, 2, 7

5 3 1 2

3 1

1

This is wrong

find min using tournament method

[6 | 2 | 3]

└ min = 2 — 2nd min

find min of compared elements = $\log n - 1$

Total $n + \log_2 n - 2$

DATE : __/__/__
PAGE : _____

The minimum number of comparisons are required to find the minimum and the maximum of 200 number is ---?

$$\frac{3n}{2} - 2 = \frac{3}{2} \times 200 - 2 = 298$$

$$\frac{3n}{2} - 2 = \frac{3}{2} \times 83 - 2 / 2$$

if the n is odd : 83

use Seal value : $3\left[\frac{n}{2}\right] - 2$

$$3 \times \frac{83}{2} - 2 = 128$$