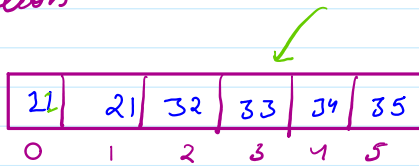


ArrayList

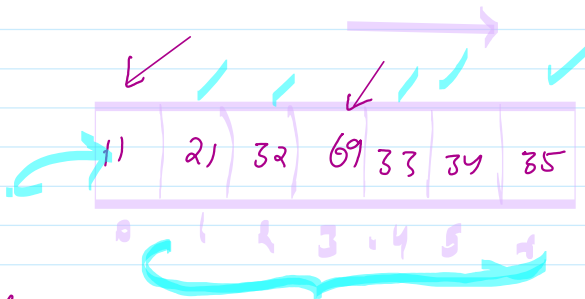
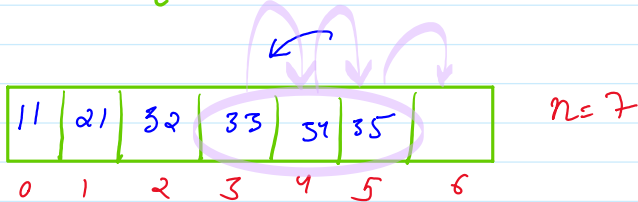
Monday, September 1, 2025 4:04 PM

Insertion



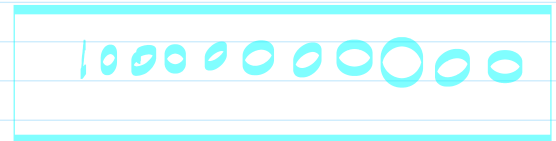
index = 3
element = 99

$n = 6$
 $arr[n+1] = 7$

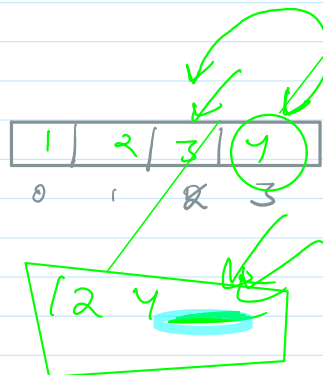


$arr[3] \Rightarrow \text{element}$

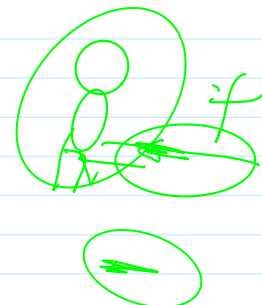
Drawback of array



Deletion:-



$Q^n =$
 $2 = \text{index}$



Arraylist:-

Arrays \rightarrow static \rightarrow \checkmark \otimes \rightarrow change

Runtime \otimes

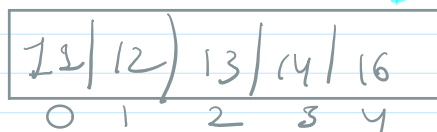
size \rightarrow increase \rightarrow runtime

Dynamic

Qⁿ How arraylist works / grows:-

Cracking the coding:-

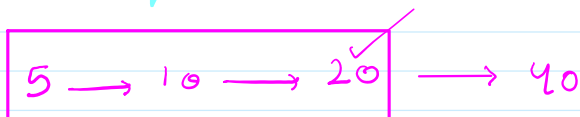
Internally array:-



+5 left=1000

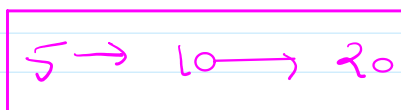


2 x previous size



5 \rightarrow 12

/



12

Methods:- → Practice
→ Remember:

Define:-

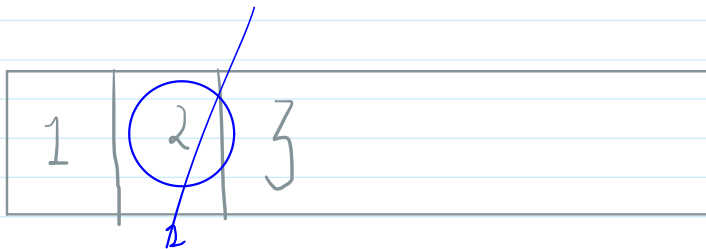
int

char

int arr[] = new int[] / ArrayList<Integer> arr = new ArrayList

add → method ✓

name.add(value)



1->2->3->4->5

int num = arr.get(index)

① add → normal
→ index

12

69

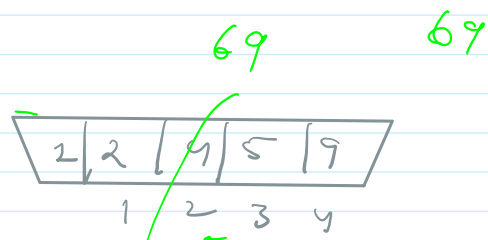
① add \rightarrow index

② remove

③ get

④ set

`arr.set(2, 69)`



arr.contains(5)

true
false

O^N

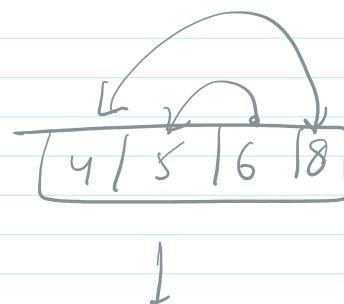
void/int/long

arr[] \rightarrow
ArrayList \rightarrow } return

f^N Array Name () {

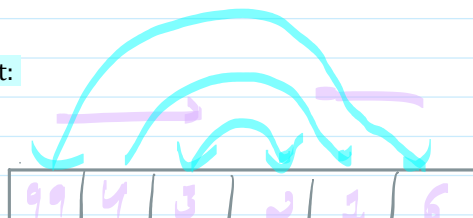
=====
}

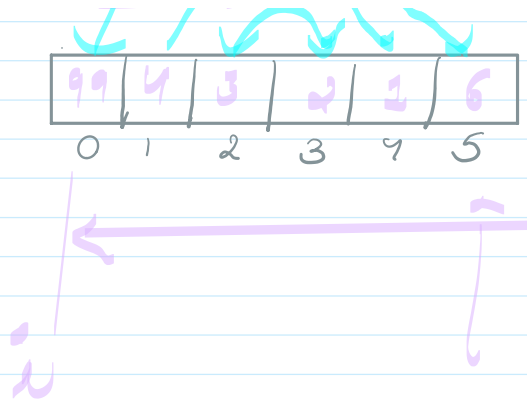
$O^N \rightarrow$



Swap of N

Reverse a numbers in arraylist:





جای

Sorting:- Arranging increasing / decreasing order

10 7 8 2 5 4 3 -1

-1 2 3 4 5 7 8 10

arraylist \rightarrow Method

collections.sort(arraylist-name)

Container with most part:-

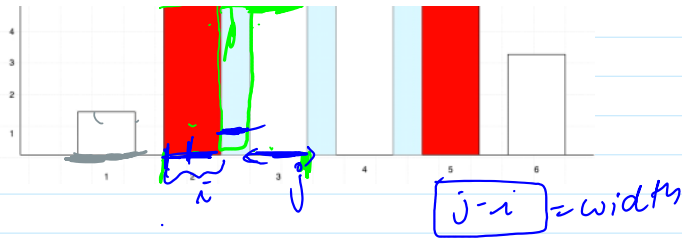


given \rightarrow height

arraylist \rightarrow 0 1 0 2

max/most water

Brute force:

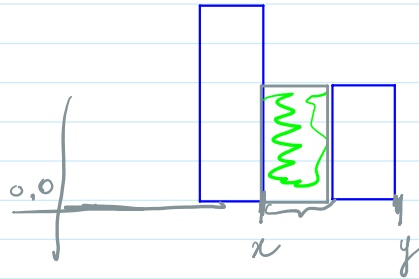


1, 8, 6, 2, 5, 4, 8, 3, 7

Brute force:

$\text{max_water} = \text{Integer.MIN_VALUE}$

$\text{max} = -\infty$



$l \times b \Rightarrow$

$y - x$