CODECLUB SMVDU

WEEK 2: BINARY SEARCH

Recommended playlist: https://www.youtube.com/playlist?list=PLgUwDviBIf0oF6QL8m22w1hIDC1vJ_BHz

Recommended problem sheet: https://takeuforward.org/strivers-a2z-dsa-course/strivers-a2z-dsa-course-sheet-2/

Problems from Coding Session:

- Linear Search
- Binary Search
- Second largest in an array
- Smallest and Second Smallest in an array
- Missing number in an array
- Missing and repeating element

Problems for practice on arrays:

- Sort an array of 0s, 1s and 2s
- Spirally traversing a matrix
- Rotate an array

Problems for practice on binary search:

- Search in rotated array
- Floor in sorted array
- Square root of a number
- Nth root of M

Lecture – 02.

Sep 11, 2023

Monday

Topics :-

1) Linear Search
2) Binary Search
3) 2nd largest ele in arr
4) 2nd Smallest ele in arr
5) Missing number
6) Duplicate numbers

for (0 → n)

{

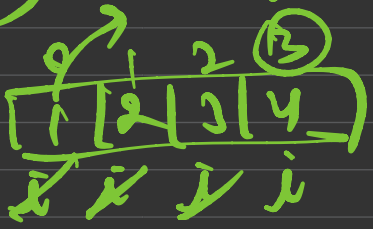
arr[i] == target

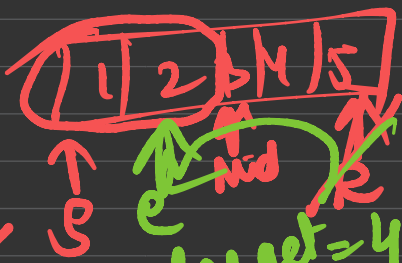
return i

}

return -1

target = 5

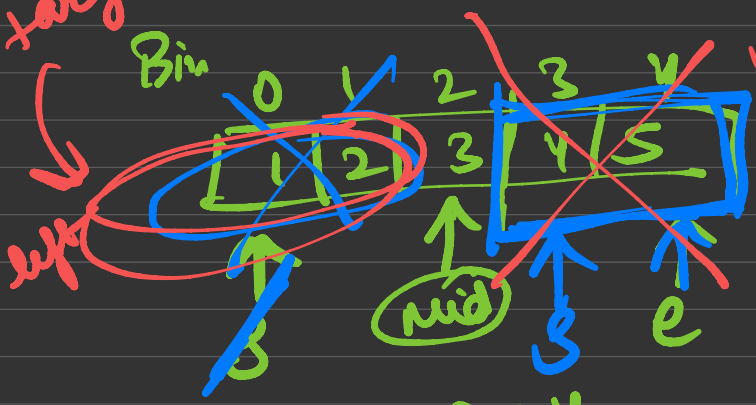




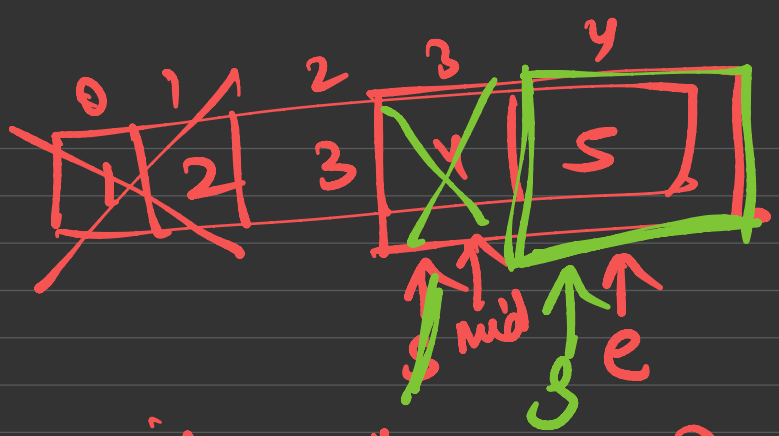
target = 4

target = 1

Bin

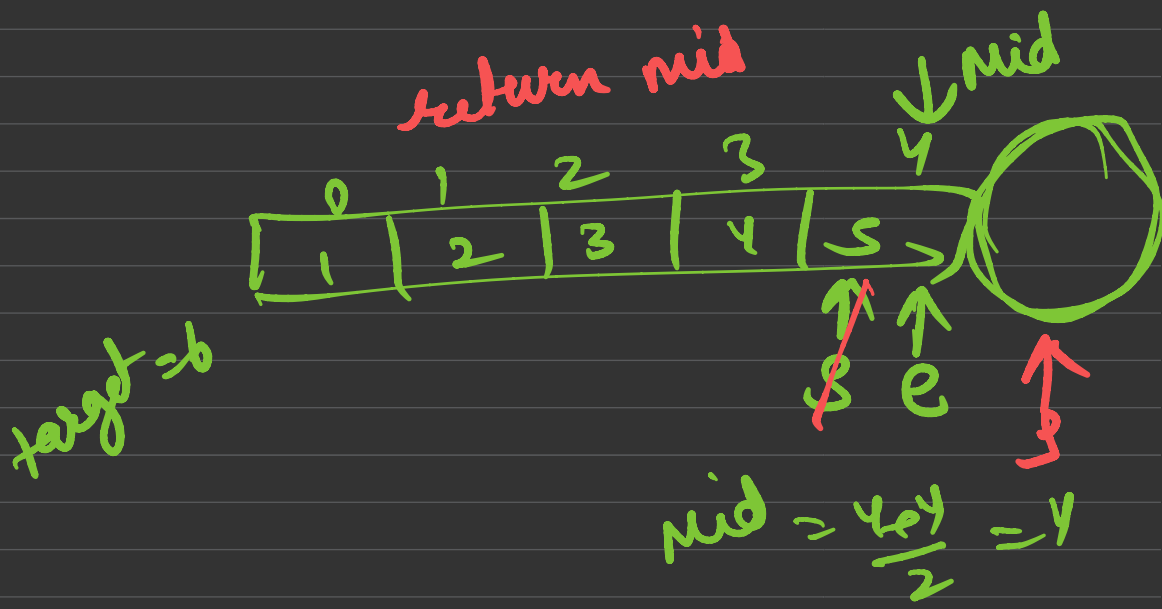


$$mid = \frac{0+4}{2} = 2$$



$$mid = \frac{3+4}{2} = 3.5 = 3$$

return mid



```
while ( s <= e)
{
    mid = (s+e)/2
    if (arr[i] == target)
        return mid
    elseif (arr[i] > target)
        e = mid-1
    else
        s = mid+1
}
return -1
```

2 largest ele

0	1	2	3	4
1	2	3	5	4

l = arr[0] = ~~1~~ ~~2~~ ~~3~~ 5

sl = ~~1~~ ~~1~~ ~~2~~ 3

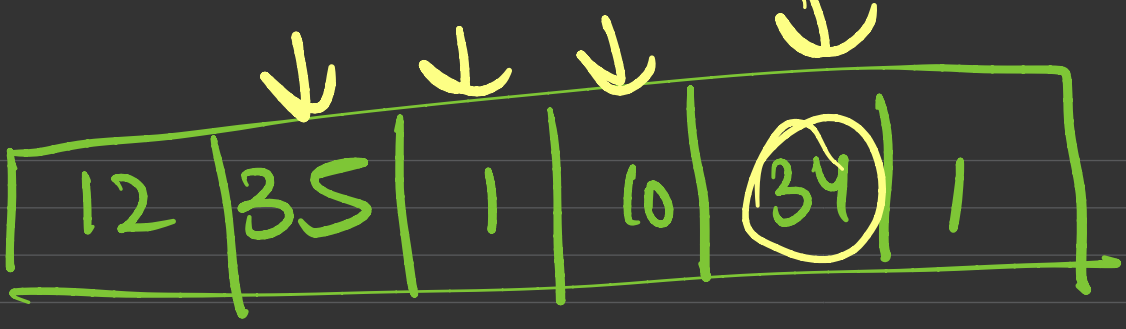
for(1 → n)
{

l

sl

arr[i]

sl



l = arr[0] ~~12~~ 35

~~sl = -1~~ ~~12~~ 34

for (1 → n)

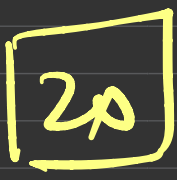
{ if (arr[i] > l)

{ sl = l

l = arr[i]

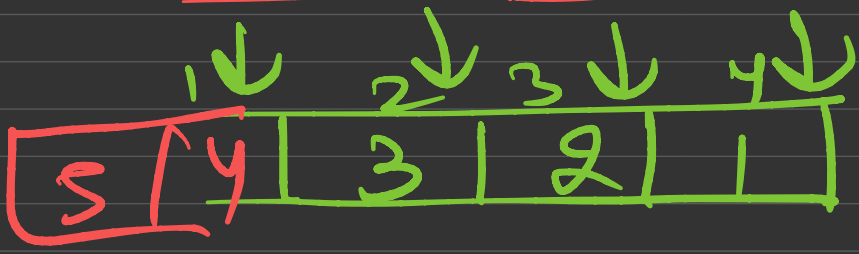
}

```
elseif(arr[i] < l
    && arr[i] > sl)
{
    sl = arr[i]
}
}
return sl
```



l = 20
sl = -1

2nd Smallest in arr



$u = arr[0] = 5\ 4\ 3\ 2\ 1$

$ss = INT\_MAX\ 5\ 4\ 3\ 2$

ss ⇒

for($1 \rightarrow n$)

< > <

3

vector<int> ans;

ans.push_back(s)

ans.push_back(ss)

return ans;

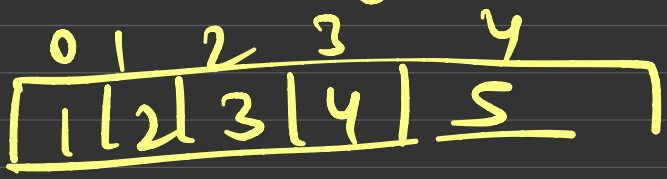
10

ss = INT_MAX

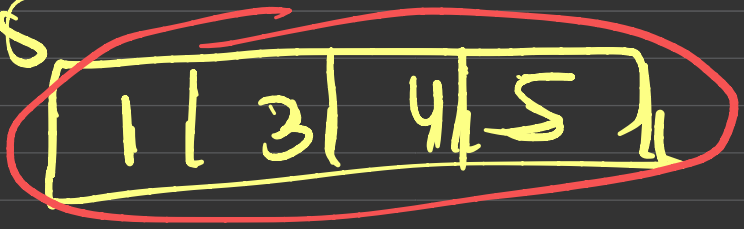
if(ss == INT_MAX)

ans.push_back(-1)

Missing number



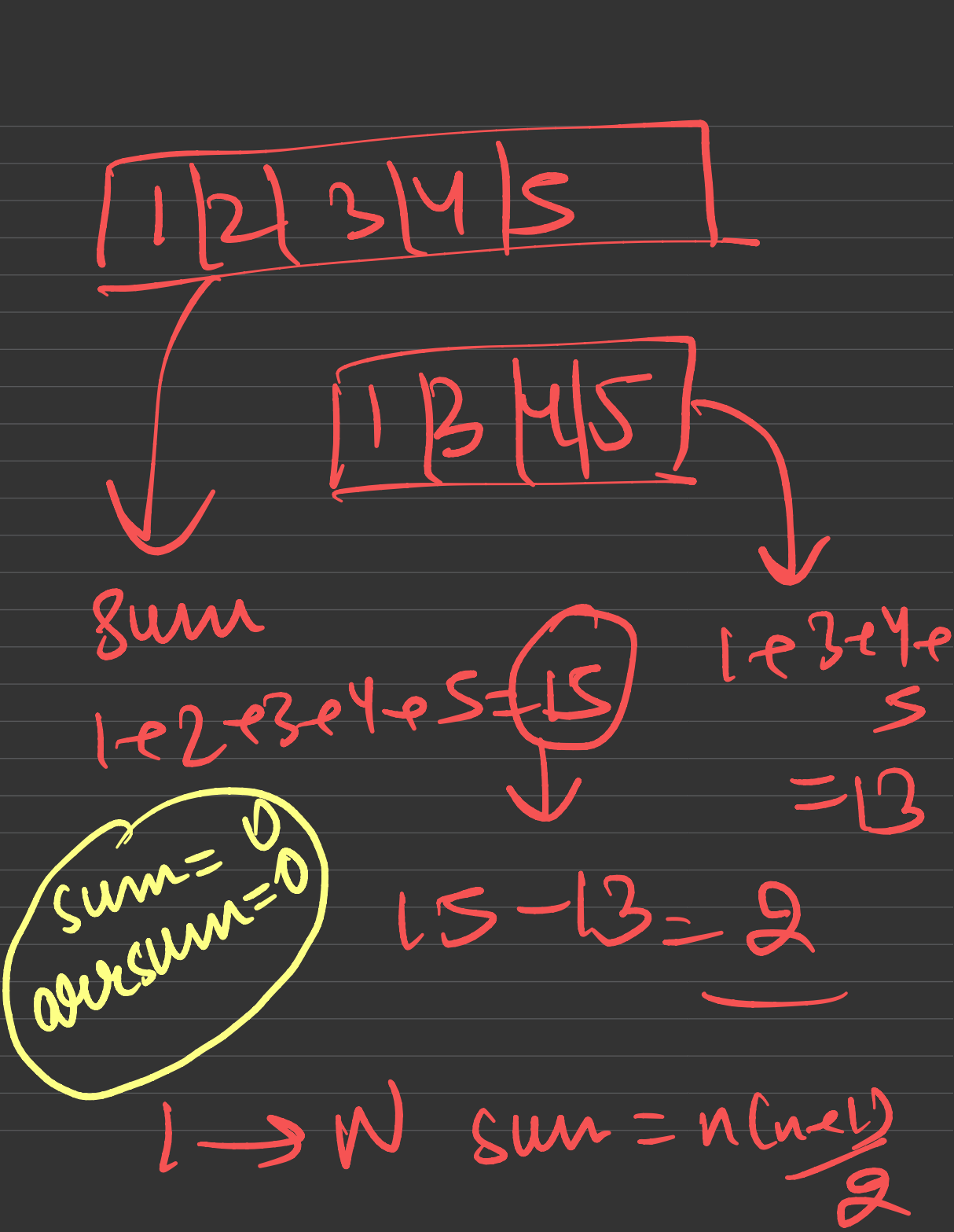
N=5



1 → N

N=5 [1 2 3 4 5]

N=10


1 2 3 4 5
1 3 4 5
sum
1+2+3+4+5=15
1+3+4+5
=13
sum=0
oursum=0
15-13=2
1 → N sum = n(n+1)/2

$sum = \frac{n(n+1)}{2}$

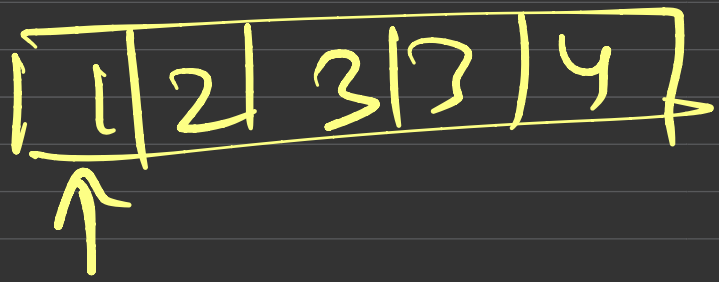
for(0 → N-1

arrsum = arrsum + arr[i]

N=5

missing = sum - arrsum

Duplicate Numbers

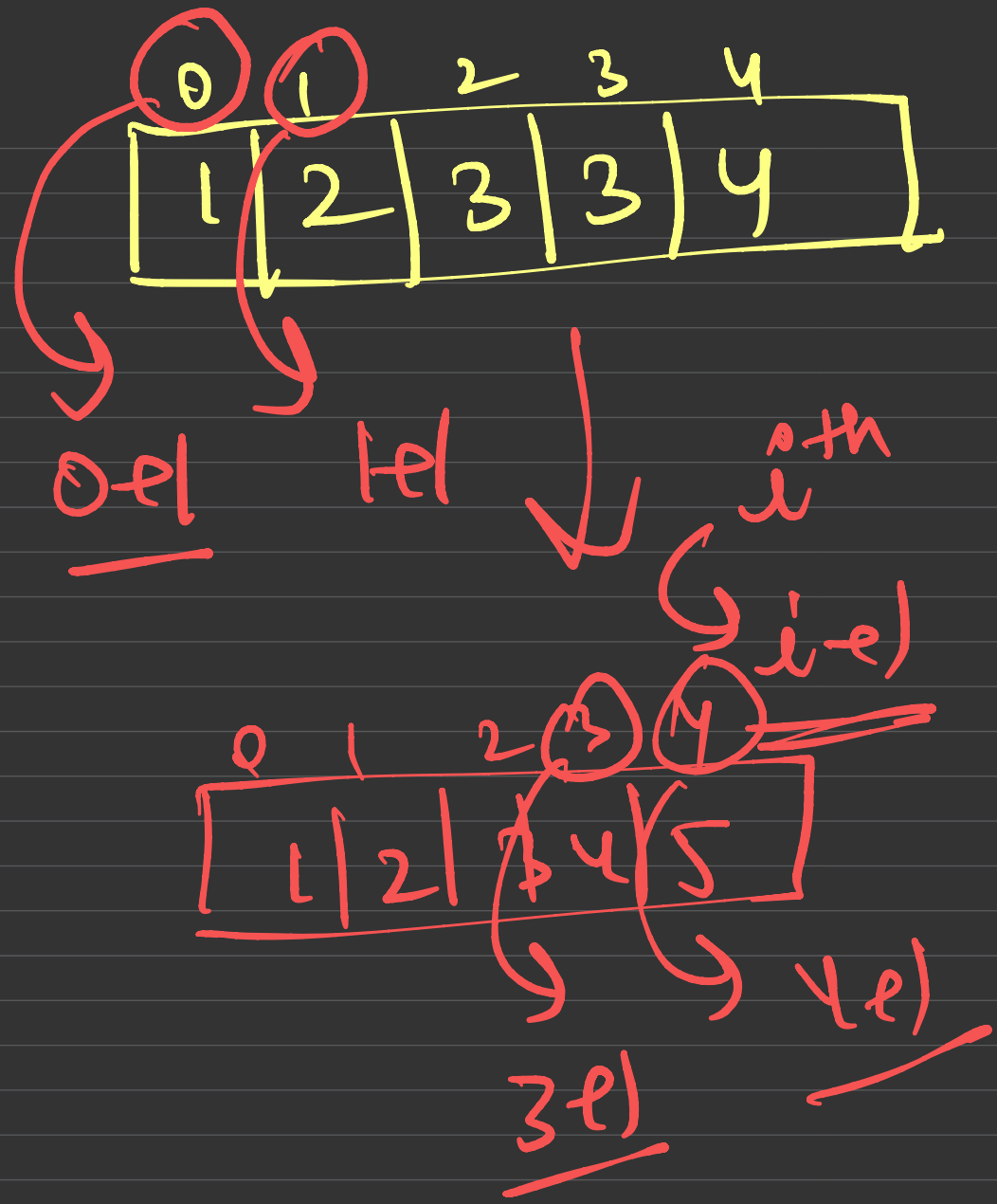


map

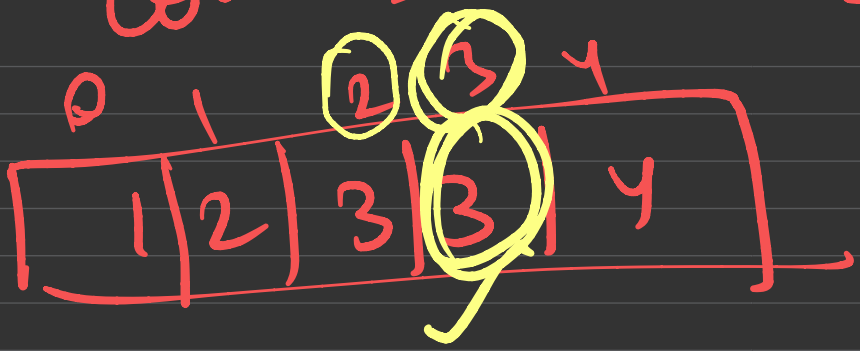
1 → 1
2 → 1
3 → 2
4 → 1

> 1

dup


0
1
2
3
4
1
2
3
3
4
0-el
1-el
i^th
i-el
0
1
2
3
4
1
2
3
4
5
3-el
4-el

condⁿ → arr sorted



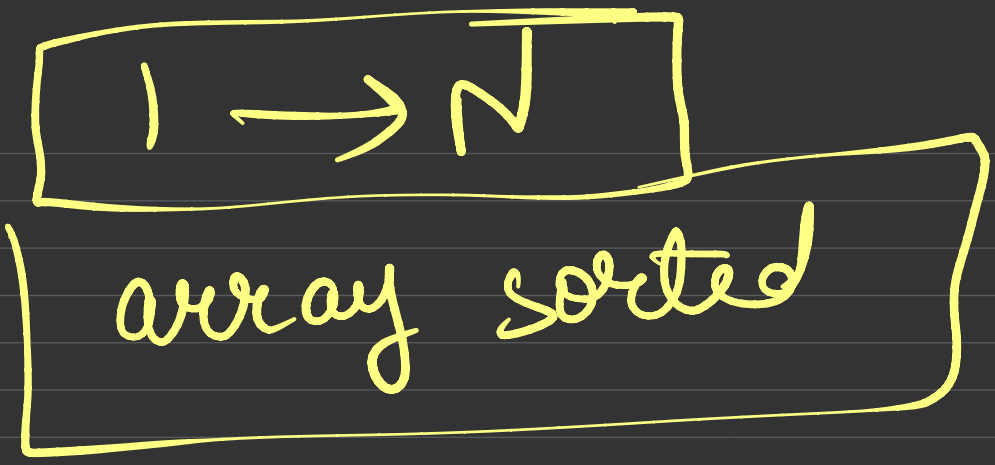
for(0 → n)

if(arr[i] - 0 != i - el)

{ dup = arr[i]

missing = i - el

}


1 → N
array sorted