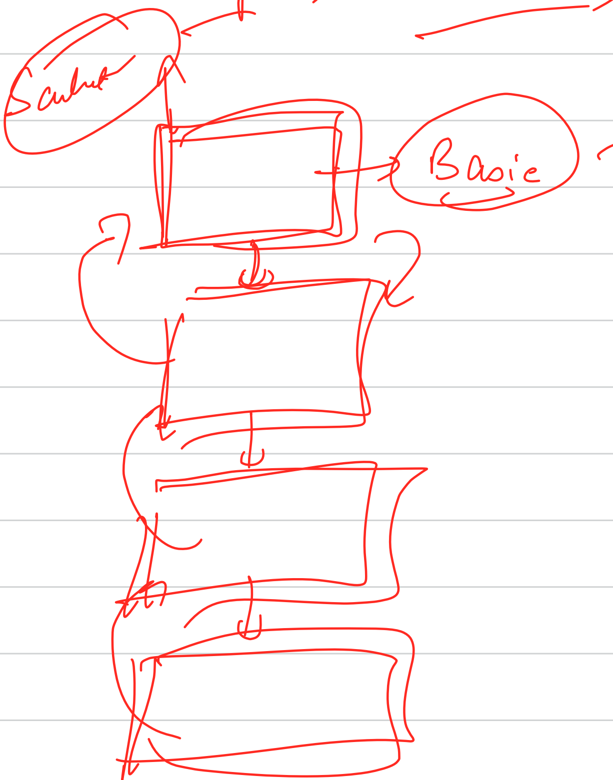
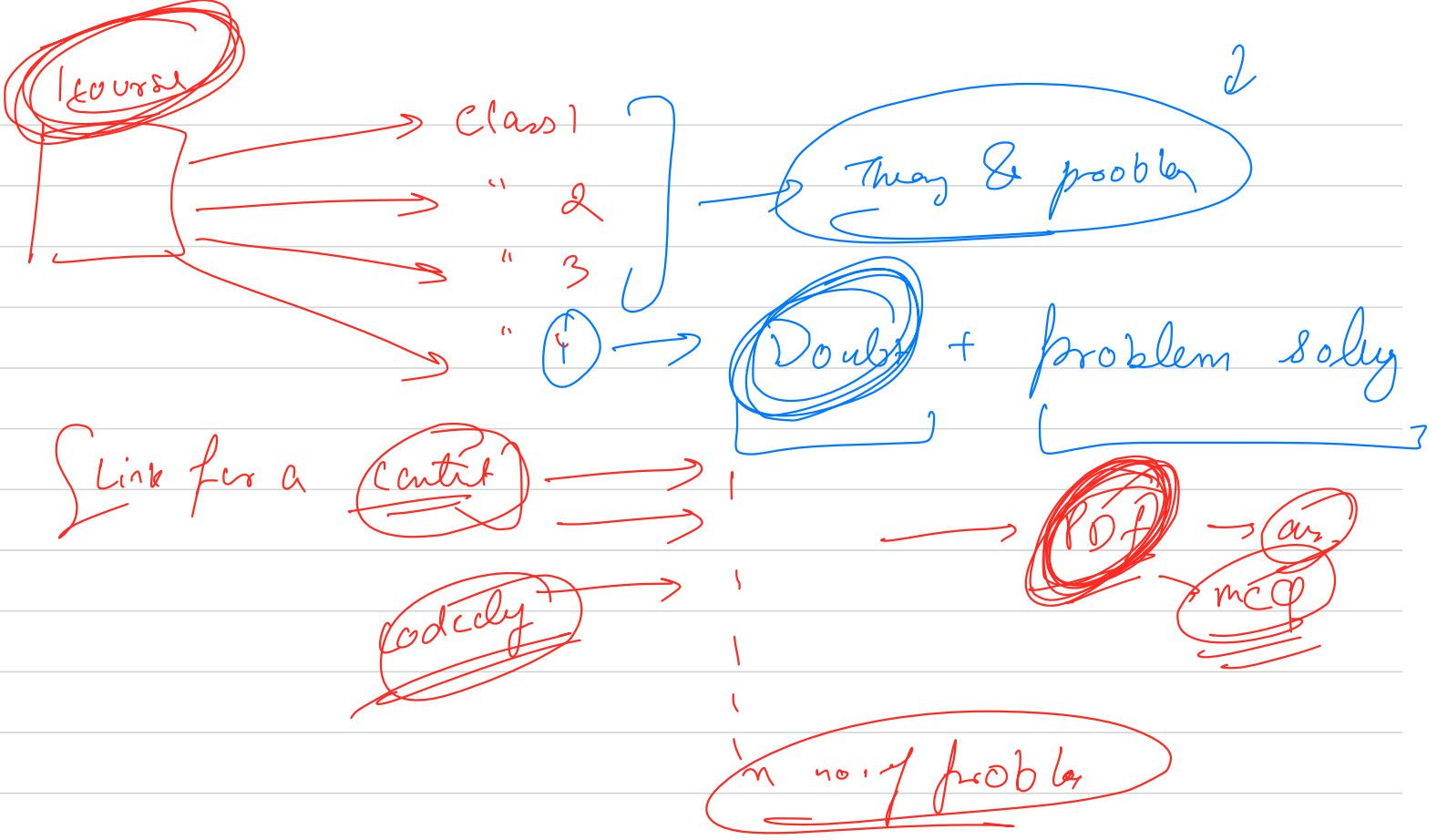
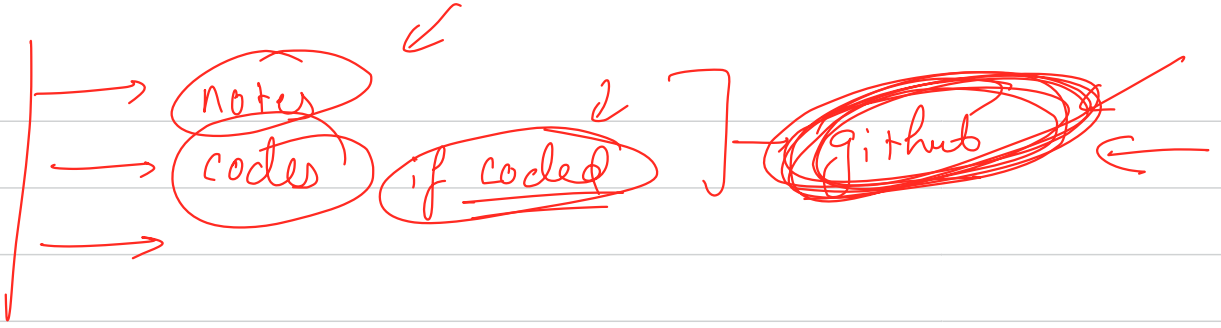
→ loops, conditionals, arrays, Recursion → prefix



vector
array
1D
2D
Fn C

Sanket
Aryan
Riya
Deepak 2


Lecture
class 1
" 2
" 3
" 4
Theory & problem
Doubt + problem solving
Link for a content
codedly
PDF
ans
mcq
n no. of problem


notes
codes
if coded
github

→ How arrays & vectors are internally implemented
→ MCQ around arrays, address, idx (C++)
→ Array will be the base for advanced dsa

↳ Q⇒ You have a vector of all zeroes. of length N.

You will get Q queries and each query has 2 values L, R. L & R represent index of the vector.

You need to increment all the indexes from [L, R] by 1.

After processing all queries, print the final vector.

↳ A = [0, 0, 0, 0, 0, 0]

Constraints

$N \leq 10^7$

$Q \leq 10^6$

$A = [0, 0, 0, 0, 0, 0]$
$0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5$

$Q \times N$

Brute force

↳ $Q = 5$

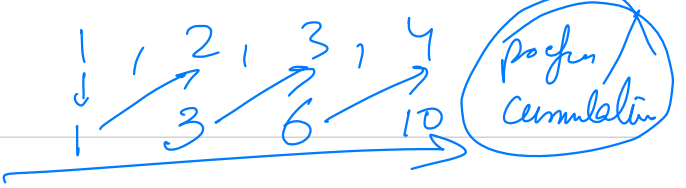
Go and process each query by manually updating the vector on each query

	L	R	
↳	2	3	$[0, 0, 1, 1, 0, 0]$
↳	2	5	$[0, 0, 2, 2, 1, 1]$
↳	0	4	$[1, 1, 3, 3, 2, 1]$
↳	3	3	$[1, 1, 3, 4, 2, 1]$
↳	3	4	$[1, 1, 3, 5, 3, 1]$

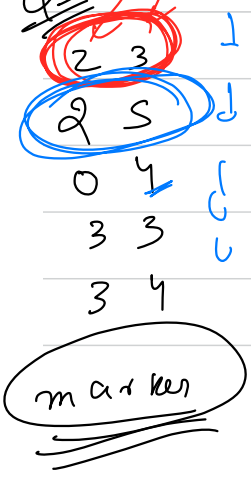
ans → $[1, 1, 3, 5, 3, 1]$

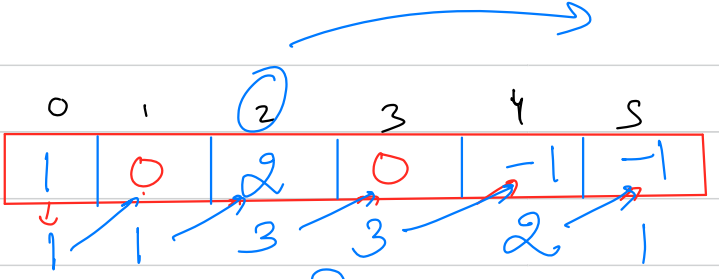
Difference vector / array

↳ How to optimise ??



↳ Anyhow we need to process the query. How about storing the queries somewhere, and change the vector at once.





$O(Q + N)$

```
a[L] += 1
a[r+1] -= 1
```

$\forall i \quad a[i] = a[i] + a[i-1]$

1 2 3 4
1 3 6 10

```
cin >> q;
while (q--) {
    cin >> l >> r;
    a[l] += 1
    if (r+1 < n)
        a[r+1] -= 1
}
for (i=1; i<n; i++) {
    a[i] = a[i] + a[i-1] ; // prefix sum
}
print(a)
```

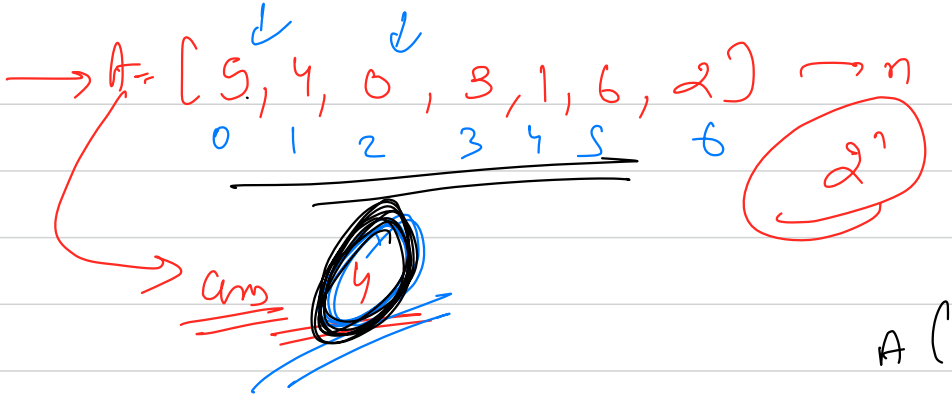
Q:- Given an array of length N, which contains elements from [0 - N-1]. Find the length of longest subset S, where

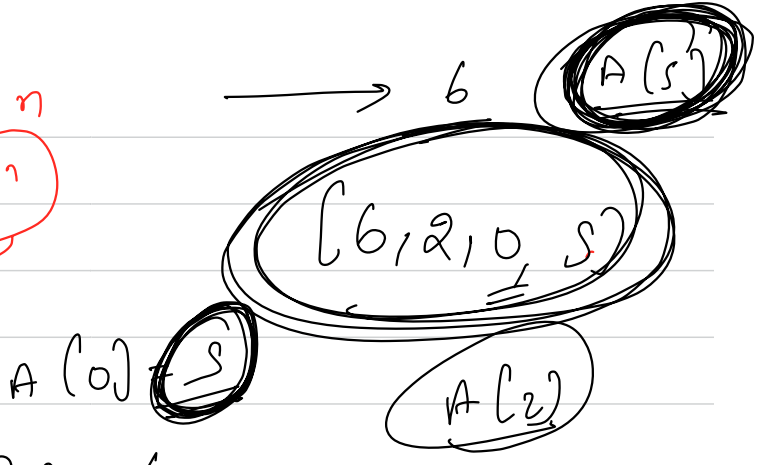
$$S[i] = \{A[i], A[A[i]], A[A[A[i]]] \ldots\}$$

Let's say first element of set is A[i], then second is found to be A[A[i]], then third will be A[A[A[i]]] & so on.

We stop when we have first duplicate found.

Return the length of largest such subset.





$A[5] = 6$

ans → [5, 6, 2, 0]

$A[0] = 5$

$A[A[0]] = A[5] = 6$

$A[A[A[0]]] = A[6] = 2$

$A[A[A[A[0]]]] = A[2] = 0$

$A[A[A[A[A[0]]]]] = A[0] = 5$ → repeated

$N \leq 10^6$

0, 5, 6, 2

⇒ [5, 4, 0, 3, 1, 6, 2]

0 1 2 3 4 5 6

[5, 6, 2, 0]

A[5]

{A[5], 2, 0, 5}

A[5] = 6

A[A[5]] = A[6] = 2

A[A[A[5]]] = A[2] = 0

A[A[A[A[5]]]] → A[0] = 5

A[A[A[A[A[5]]]]] = A[5] = 6 → ans

A[4] = 1

{1, 4} → 2

A[A[4]] = A[1] = 4

A[A[A[4]]] = A[4] = 1

TC → O(n)

SC → O(1)

Vector → push_back → O(1)

array → add a elmt at last

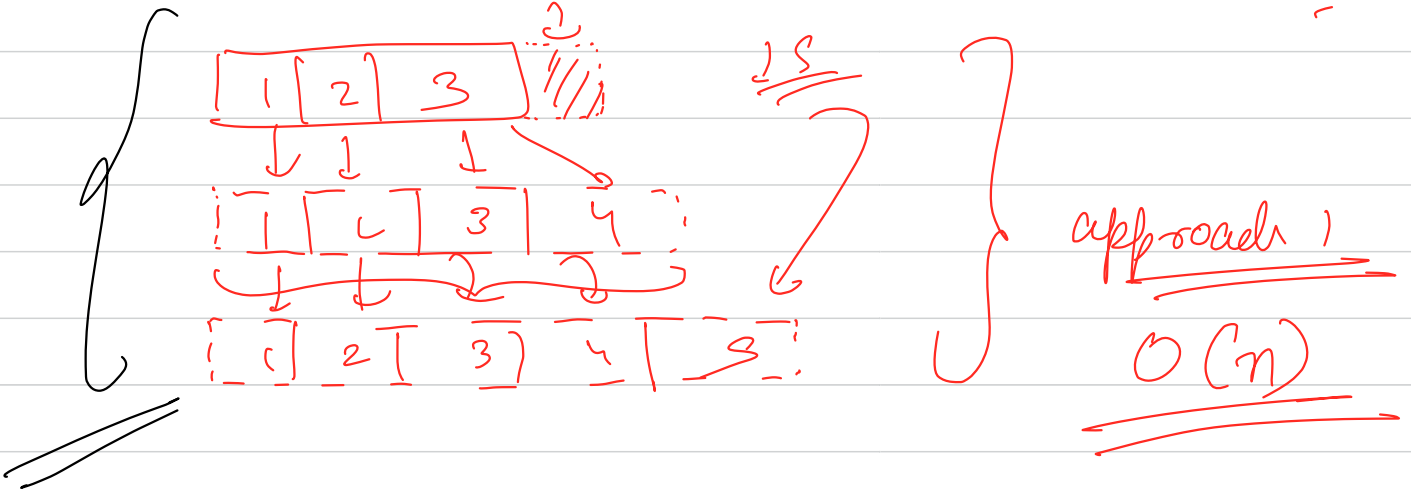
access

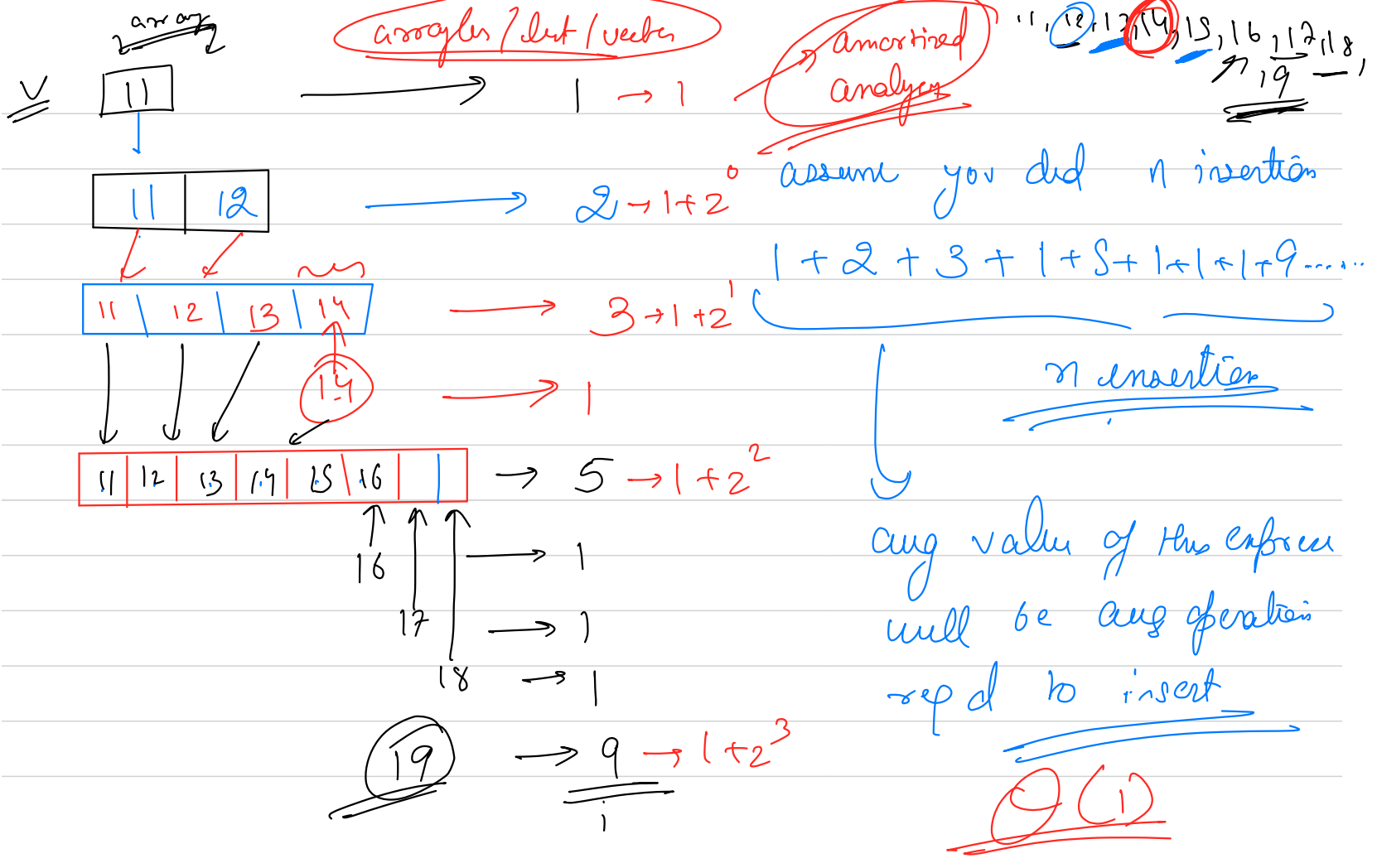
no

O(n)

O(1)

How vector maintains const time insertion




array
11
12
13
14
15
16
17
18
19
arraylist / dot / vector
amortized analysis
1 → 1
2 → 1+2^0
3 → 1+2^1
1
5 → 1+2^2
1
1
1
9 → 1+2^3
1
assume you did n insertion
1+2+3+1+5+1+1+1+9.....
n insertion
avg value of the expression will be avg operation req'd to insert
O(1)

$$\frac{1+2+3+1+5+1+1+1+9\cdots\cdots\cdots}{n}$$

$$\frac{1+(1+2^0)+(1+2^1)+1+(1+2^2)+1+1+1+(1+2^3)\cdots}{n}$$

$$\frac{\overbrace{1+1+1+1\cdots\cdots}^{n} + \overbrace{2^0+2^1+2^2\cdots\cdots}^{\log_2 n}}{n}$$

$$\frac{n + \frac{1\times(2^{\log_2 n}-1)}{2-1}}{n} = \frac{n+n-1}{n}$$

Cancel

→ C++ specific for vectors

↳ By default if you pass vector in a func" it is passed By copy. Make sure unless reqm, pass it by reference.