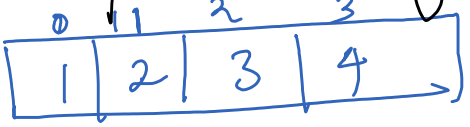
Session 3

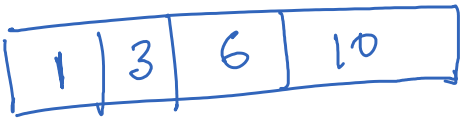
28 July 2021 09:51

* Running sum of a 1D array.

Input:



Output:



Approach:

```
int sum = 0;
for (i=0; i<n; i++)
{
    sum = sum + A[i]
    A[i] = sum;
}
```

3

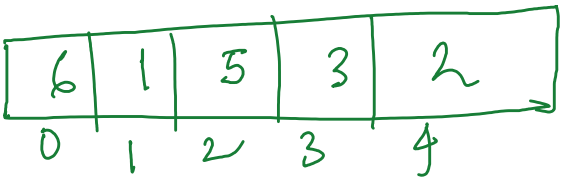
* Find out the missing element

Given an array having n elements
All elements are unique — no duplicates.
The range of the no.s is 1 to N

$N = n+1$

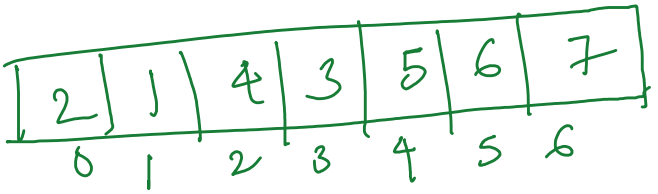
n=5
N=6
1–6

Input:



Output: 4

Example 2:



n=7
N=8.
1 to 8

Output: 8

Approach 1 : Sorting:

n=5,
1 to 6.

Approach 1:

Input.

6	1	5	3	2

n=5, 1 to 6.

Sorted

1	2	3	5	6

↑ i = 1 to 5

i =

Time complexity: O(nlogn)

Space complexity: O(1)

1 to N

Approach 2:

Sum of first N natural no.s = $\frac{n*(n+1)}{2}$

6	1	5	3	2

n=5.

1 to N

1 to 6

N = n+1

1 2 3 4 5 6

(1+2+3+4+5+6) − (6+1+5+3+2)

= 4

```
int N = n+1;
int sumN = (N*(N+1))/2, sum=0;
for(i=0; i<n; i++)
{   sum = sum + A[i];
}
int result = sumN - sum
```

Height Checker

count = 3

Input :

expected :