Arrays in C++ (9.1)

An Array is a data structure used to store blocks of information in contiguous memory allocation. The data can be integer, strings, characters, class objects, etc.

Declaring an Array:

```
//Syntax
//<data_type> Name[size]
int intArray[100];
char characterArray[100];
```

Getting started with Arrays:

1. Finding Max and Min element:
   Step 1: Taking the input 'n' and declaring an Integer Array of size 'n.'
   Step 2: Iterating the Array and calculating the maximum and minimum elements.

```
int n;
cin >> n;

int arr[n];
for (int i = 0; i < n; i++) {
    cin >> arr[i];
}

int maxNo = INT_MIN;
int minNo = INT_MAX;

for (int i = 0; i < n; i++) {
    maxNo = max(maxNo, arr[i]);
    minNo = min(minNo, arr[i]);
}

cout << maxNo << " " << minNo << endl;
```

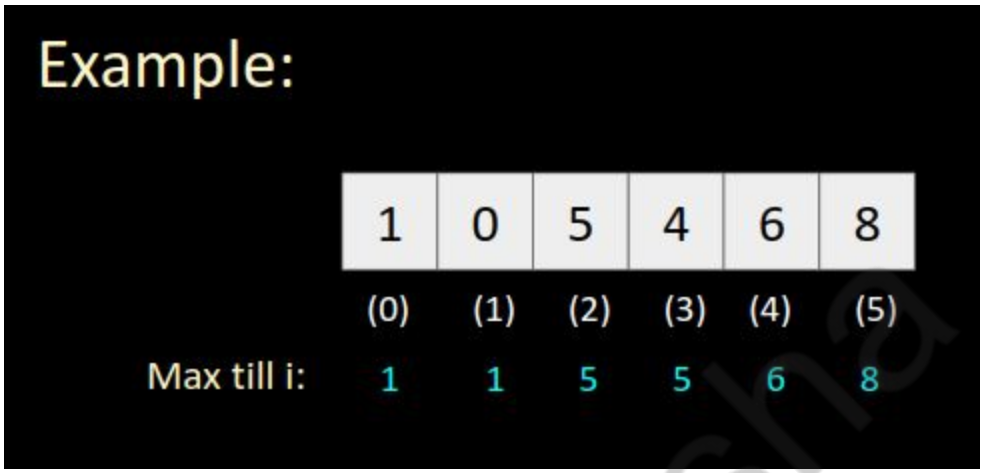
Practice questions:

1. Running sum of an Array
2. Kids with the greatest number of candies

Arrays Challenge (Max till i)

Problem:

Given an array a[] of size n. For every i from 0 to n-1 output max(a[0], a[1],..., a[i]).

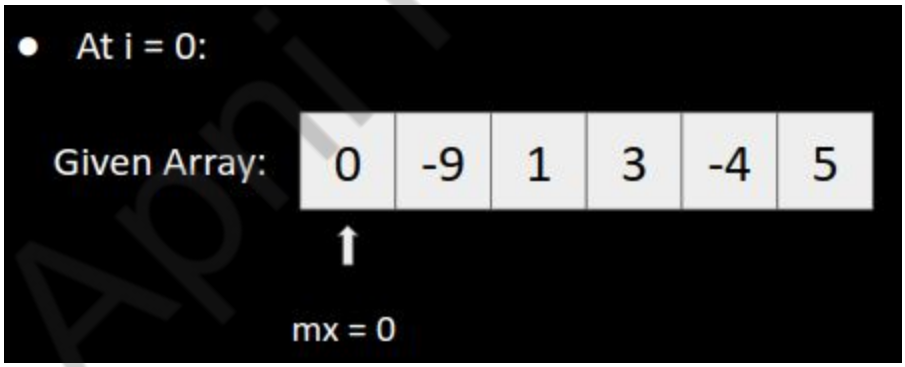


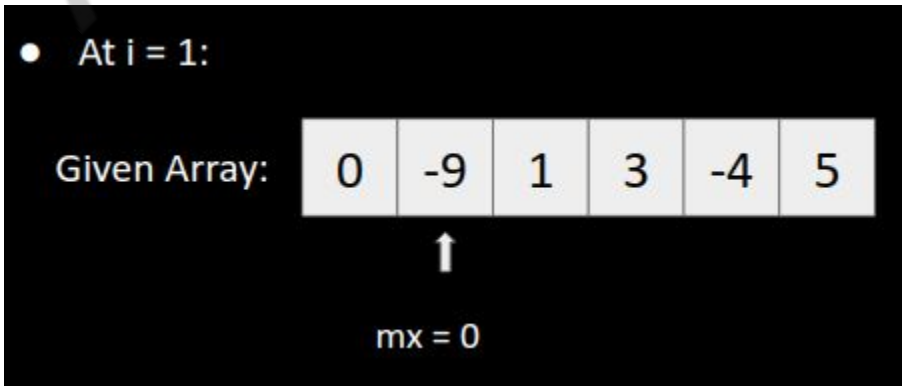
Approach:

1. Keep a variable mx which stores the maximum till i^{th} element.
2. Iterate over the array and update,

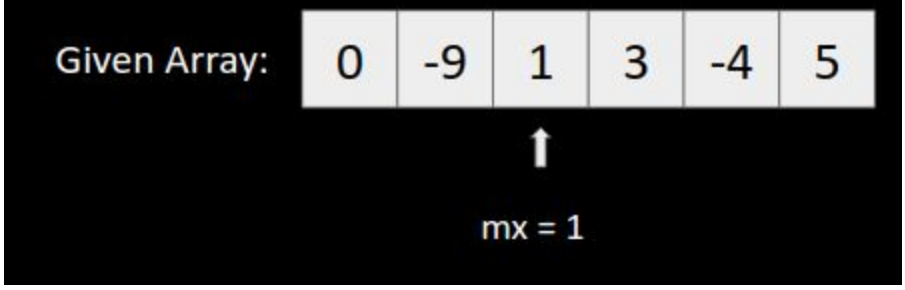
mx = max(mx, a[i])

Iterations:

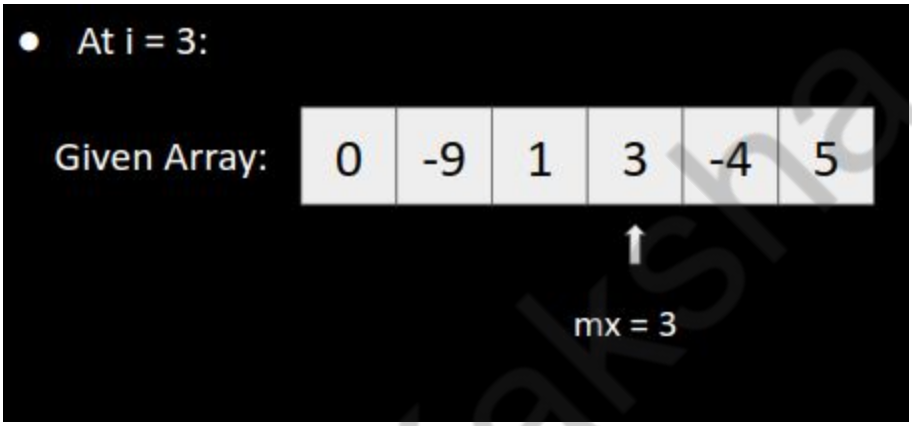




- At i = 2:

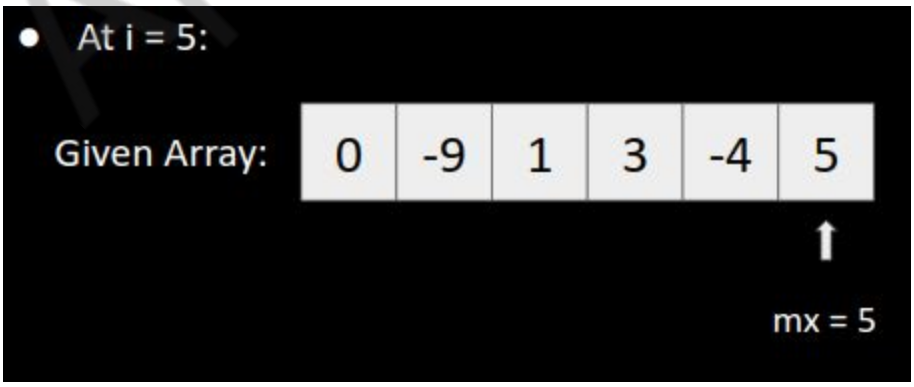


- At i = 3:



- At i = 4:

- At i = 5:



Code:

```cpp
int main()
{
    int n;
    cin >> n;

    int a[n];
    for(int i=0; i<n; i++)
    {
        cin >> a[i];
    }

    int mx = -199999;
    for(int i=0; i<n; i++)
    {
        mx = max(mx, a[i]);
        cout << mx << endl;
    }
    return 0;
}
```

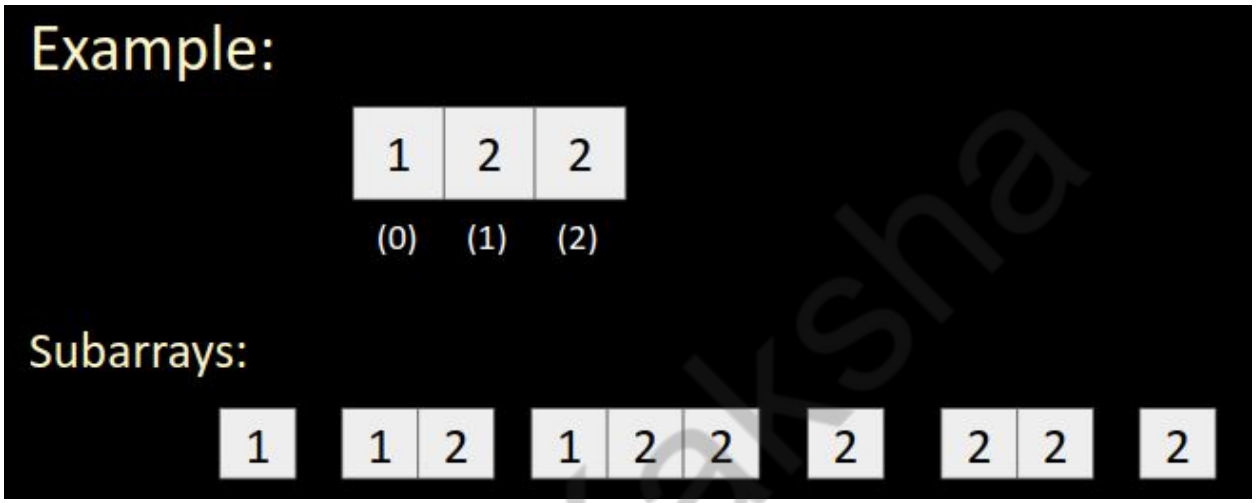
Time Complexity: O(n).

Subarray v/s Subsequence

Subarray

Subarray is a continuous part of the array.

Note: Number of subarrays of an array with n elements = ${}^{n}C_{2} + n = n*(n+1) / 2$.



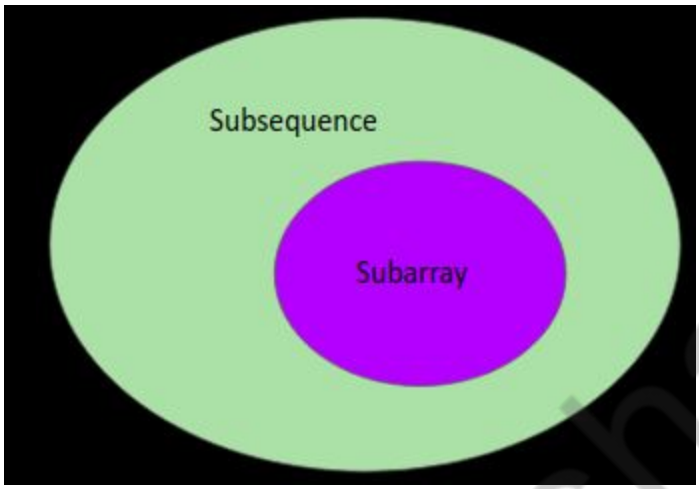
Subsequence

A subsequence is a sequence that can be derived from an array by selecting zero or more elements, without changing the order of the remaining elements.

Note: Number of subsequences of an array with n elements = 2^{n}.

Subsequences of the above array are: {}, {1}, {2}, {2}, {1,2}, {1,2}, {2,2}, {1,2,2}.

Remember: Every subarray is a subsequence but every subsequence is not a subarray.



Arrays Challenge-Longest Arithmetic Subarray (Google kickstart)

Problem

An arithmetic array is an array that contains at least two integers and the differences between consecutive integers are equal. For example, [9, 10], [3, 3, 3], and [9, 7, 5, 3] are arithmetic arrays, while [1, 3, 3, 7], [2, 1, 2], and [1, 2, 4] are not arithmetic arrays.

Sarasvati has an array of **N** non-negative integers. The i-th integer of the array is $\mathbf{A}_i$. She wants to choose a contiguous arithmetic subarray from her array that has the maximum length. Please help her to determine the length of the longest contiguous arithmetic subarray.

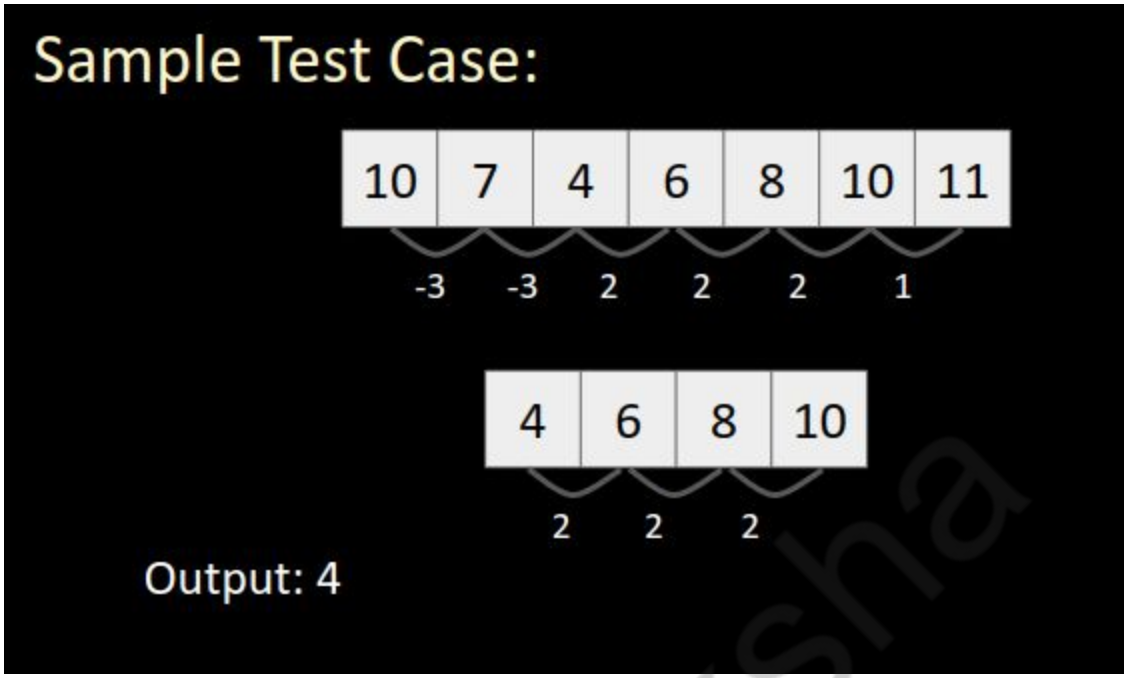
Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each test case begins with a line containing the integer **N**. The second line contains **N** integers. The i-th integer is $\mathbf{A}_i$.

Output

For each test case, output one line containing Case #x: y, where x is the test case number (starting from 1) and y is the length of the longest contiguous arithmetic subarray.

Constraints
Time limit: 20 seconds per test set.
Memory limit: 1GB.
$1 \leq \mathbf{T} \leq 100$.
$0 \leq \mathbf{A}_i \leq 10^9$.
Test Set 1
$2 \leq \mathbf{N} \leq 2000$.
Test Set 2
$2 \leq \mathbf{N} \leq 2 \times 10^5$ for at most 10 test cases.
For the remaining cases, $2 \leq \mathbf{N} \leq 2000$.



Solution

Constraints Analysis

1 sec = 10^8 operations

20 sec = $2x10^9$ operations

Intuition: We have to loop over the array and find the answer.

Steps

1. While iterating in the array we need to maintain the following variables,
    a. Previous common difference (pd) - To compare it with current common difference (a[i] - a[i-1]).
    b. Current arithmetic subarray length (curr) - It denotes the arithmetic subarray length including a[i].
    c. Maximum arithmetic subarray length (ans) - It denotes the max. Arithmetic subarray length till a[i].
2. While iterating, there will be two cases,
    a. pd = a[i] - a[i-1]
        i. Increase curr by 1
        ii. ans = max(ans, curr)
3. After loop ends, output the answer. (stored in ans variable).

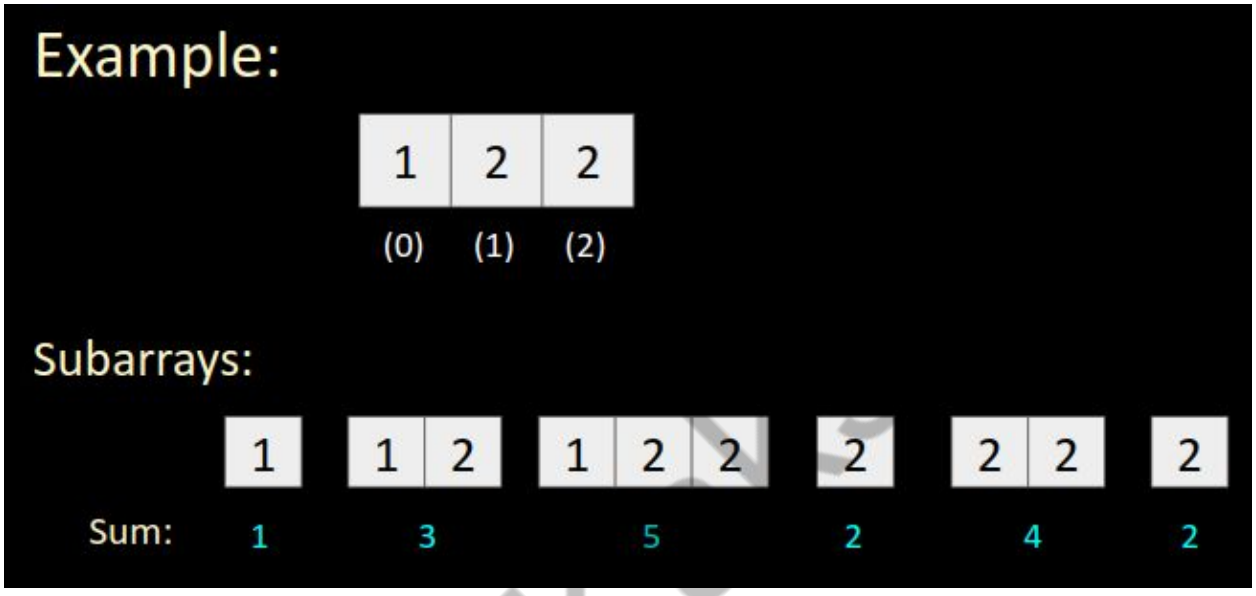
Code

```cpp
int main()
{
    int n;
    cin >> n;

    int a[n];

    for(int i=0; i<n; i++)
        cin >> a[i];

    int ans = 2;
    int d = a[1]-a[0];
    int j=2;
    int curr=2;
    while(j<n)
    {
        if(a[j]-a[j-1] == d)
            curr++;
        else
        {
            d = a[j]-a[j-1];
            curr=2;
        }
        ans = max(ans,curr);
        j++;
    }
    cout << ans << endl;
    return 0;
}
```

Arrays Challenge-Sum of all Subarrays

Question

Given an array a[] of size n. Output sum of each subarray of the given array.

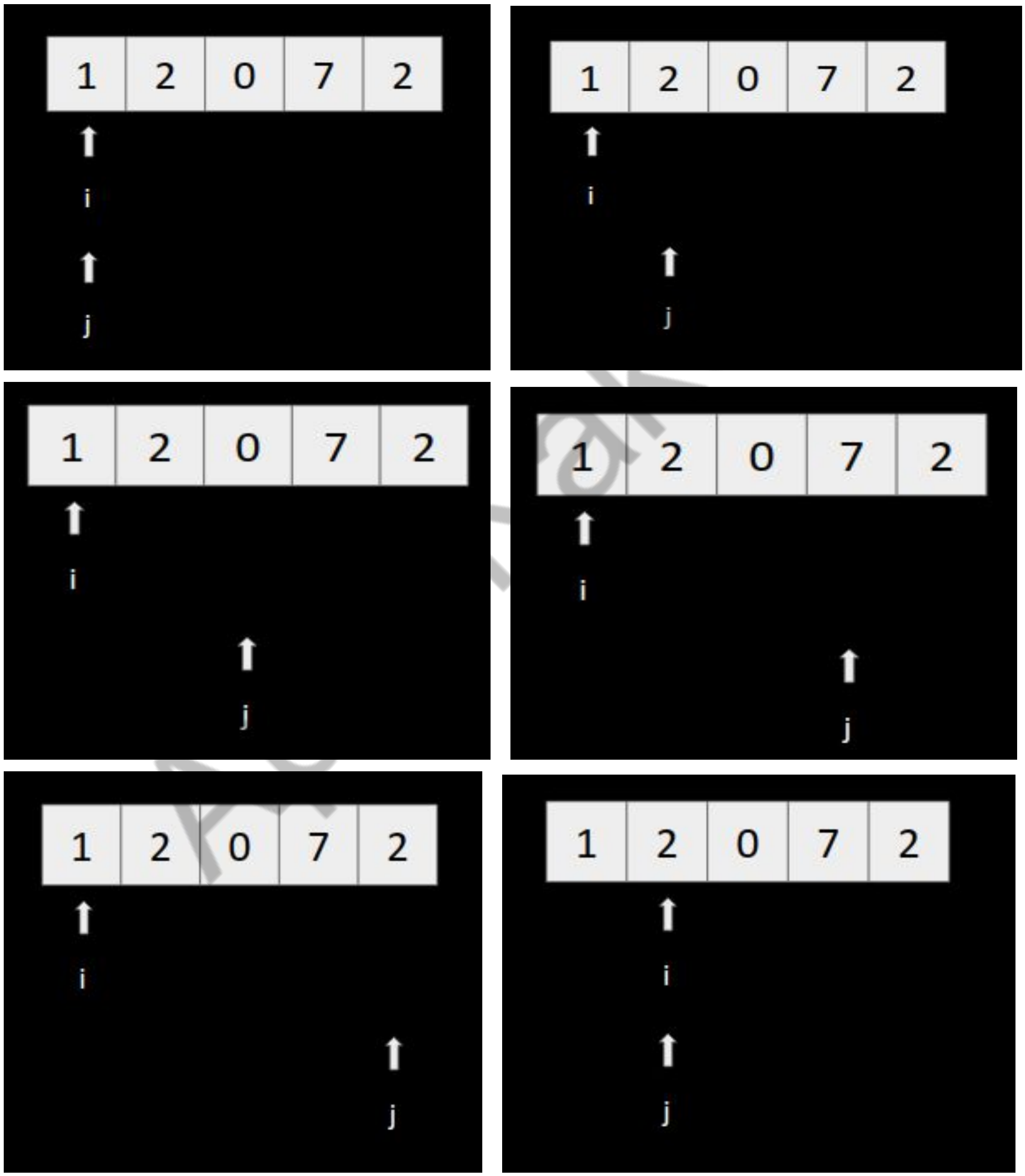


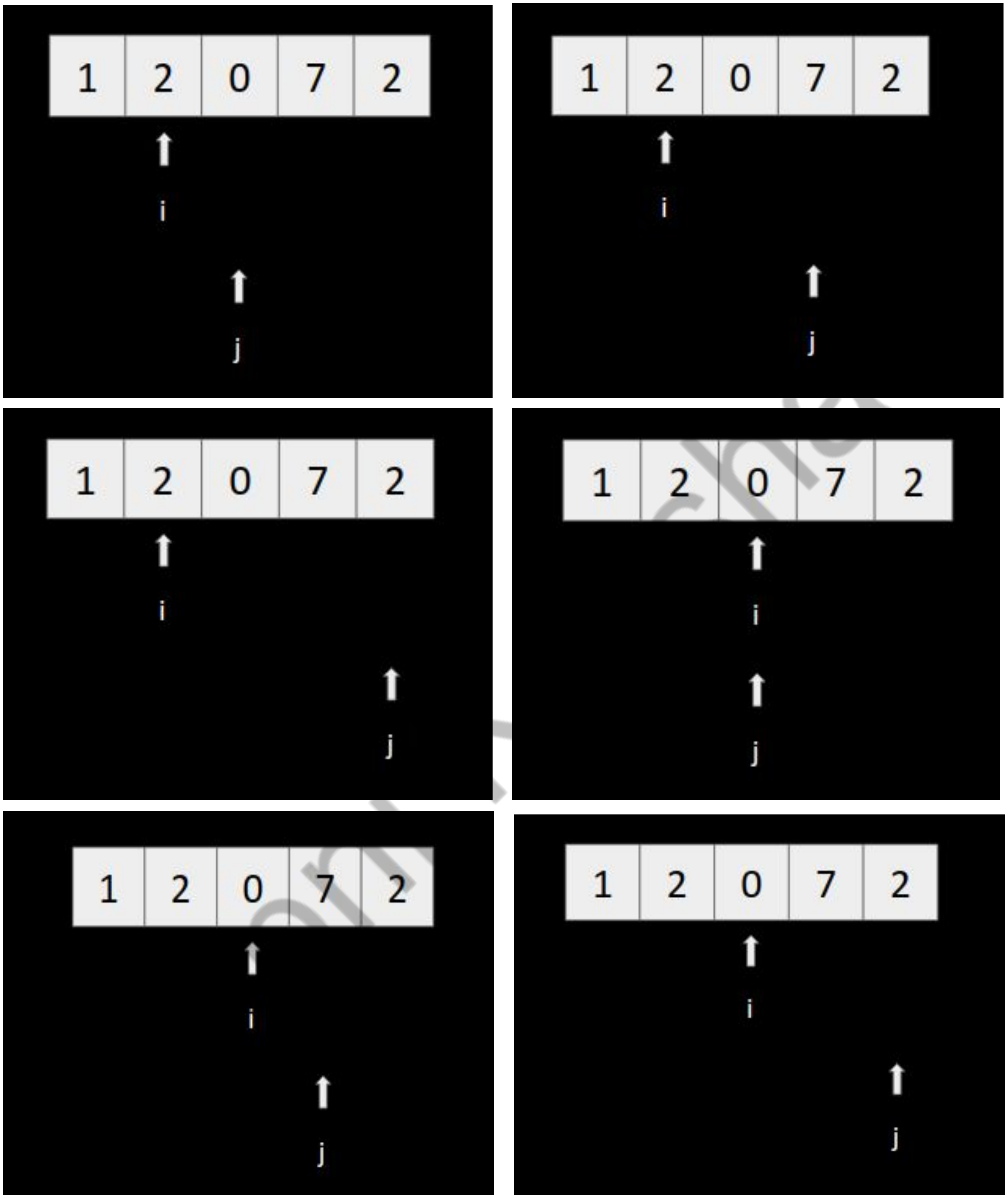
Idea: Iterate over all subarrays and output the sum after each iteration.

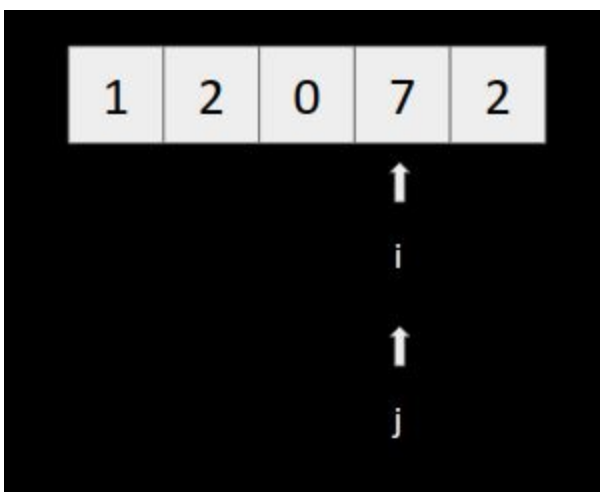
Approach

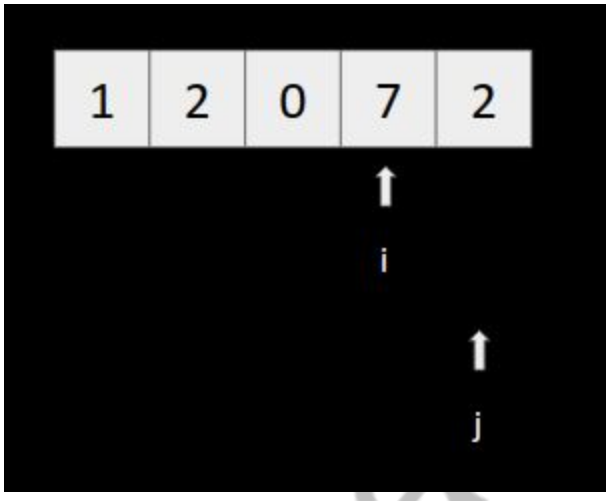
1. Write a nested loop, where outer loop runs from i=0 to i=n-1 and inner loop runs from j=i to j=n-1.
2. Keep a curr variable which stores the sum from i to j.
3. Output curr after each iteration of inner loop. After inner loop ends, update curr = 0.

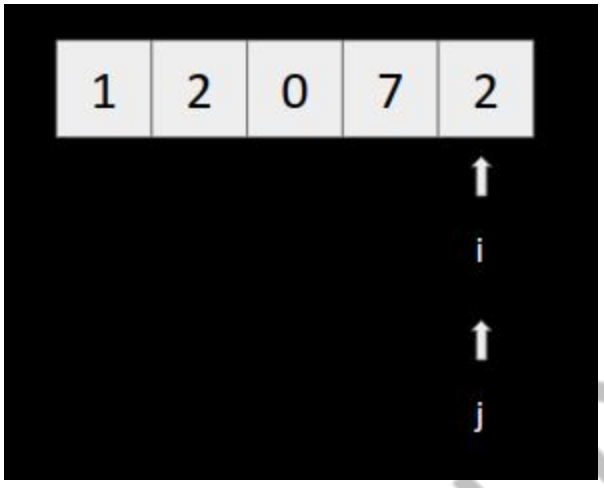
Dry Run











This question tells us how to iterate over all the subarrays. This idea is very useful in many questions which deals with operations on subarrays.

Code

```
void sumOfAllSubarrays()
{
    int n;
    cin >> n;

    int a[n];

    for(int i=0; i<n; i++)
        cin >> a[i];

    int curr=0;
    for(int i=0; i<n; i++){
        curr = 0;
        for(int j=i; j<n; j++){
            curr += a[j];
            cout << curr << endl;
        }
    }
}
```

Arrays Challenge-Record Breaker
(Google kickstart)

Problem

Isyana is given the number of visitors at her local theme park on **N** consecutive days. The number of visitors on the i-th day is $\mathbf{V_i}$. A day is *record breaking* if it satisfies both of the following conditions:

- The number of visitors on the day is strictly larger than the number of visitors on each of the previous days.
- Either it is the last day, or the number of visitors on the day is strictly larger than the number of visitors on the following day.

Note that the very first day could be a record breaking day!

Please help Isyana find out the number of record breaking days.

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each test case begins with a line containing the integer **N**. The second line contains **N** integers. The i-th integer is $\mathbf{V_i}$.

Output

For each test case, output one line containing Case #x: y, where x is the test case number (starting from 1) and y is the number of record breaking days.

Constraints

Time limit: 20 seconds per test set.

Memory limit: 1GB.

$1 \leq \mathbf{T} \leq 100$.

$0 \leq \mathbf{V_i} \leq 2 \times 10^5$.

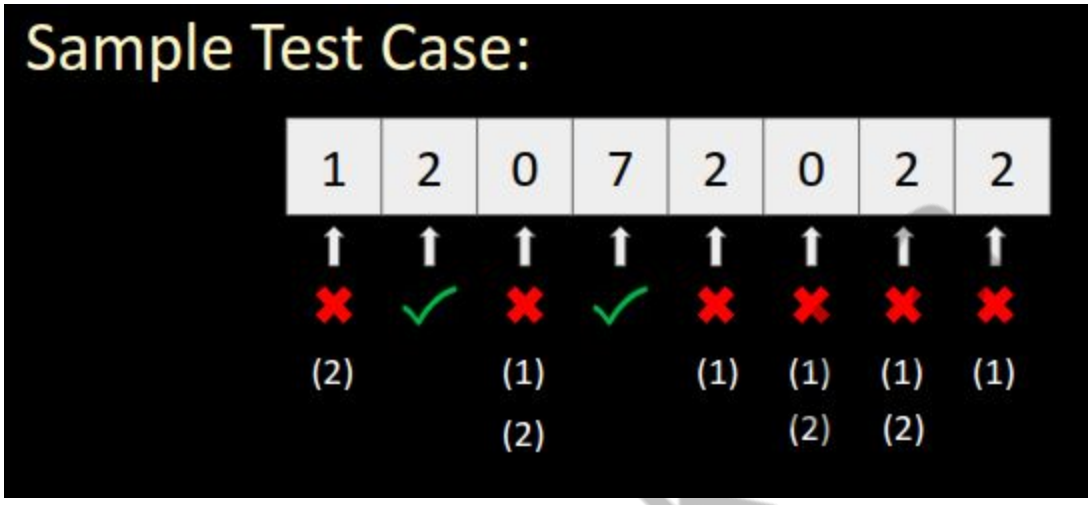
Test set 1

$1 \leq \mathbf{N} \leq 1000$.

Test set 2

$1 \leq \mathbf{N} \leq 2 \times 10^5$ for at most 10 test cases.

For the remaining cases, $1 \leq \mathbf{N} \leq 1000$.



Solution

Constraints Analysis

1 sec = 10^8 operations

20 sec = $2x10^9$ operations

Brute Force Approach

Iterate over all the elements and check if it is record breaking day or not.

Note: To check if a[i] is a record breaking day, we have to iterate over a[0], a[1],...., a[i-1].

Time complexity for this operation: O(n)

Overall Time Complexity: $O(n^2)$

Optimised Approach

Intuition: If we can optimise step (1), then we can optimise our overall solution.

For step (1): We need to check if a[i] > { a[i-1], a[i-2],..., a[0] }, which is same as

$$a[i] > \max(a[i-1], a[i-2], \ldots, a[0])$$

For this, we will keep a variable mx, which will store the maximum value till a[i]. Then we just need to check,

$$a[i] > mx$$

$$a[i] > a[i+1] \text{ , } \{ \text{ if } i+1 < n \}$$

and update mx, $mx = \max(mx, a[i])$

So step (1) time complexity reduces to O(1).

Overall time complexity: O(n)

Code

```
void RecordBreakers()
{
    int n;
    cin >> n;
    int a[n+1];

    for(int i=0; i<n; i++)
    {
        cin >> a[i];
    }
    a[n] = -1;
    if(n == 1)
    {
        cout << "1" << endl;
        return;
    }
    int ans = 0;

    int mx=-1;

    for(int i=0; i<n; i++)
    {
        if(a[i]>mx && a[i]>a[i+1])
            ans++;
        mx = max(mx,a[i]);
    }
    cout << ans << endl;
}
```