✓ 1. Alternating subarrays (Subarrays)

✓ 2. Count increasing triplets (Interview Problems 1) AS Q3

✓ 3. Christmas Trees (Interview Problems 1) HW Q2

✓ 4. Pick from both sides. (carry forward HW Q8)

Q. An array of size N is given. int B given.
↓
binary.

How many indices can act as a centre of 2B+1 length alternating array.

0-1 alternating array → 010101010

101010

10110

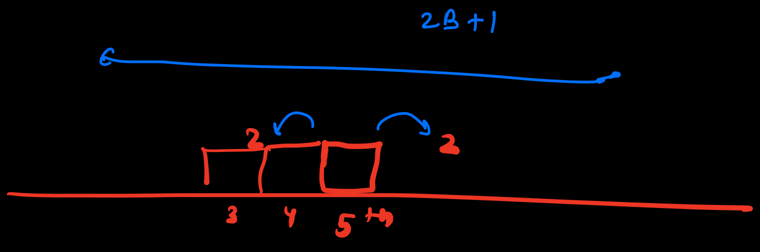
00001111

A = [1, 0, 1, 0, 1] (indices 0 1 2 3 4)

B = [1]

ans = { 1, 2, 3 }

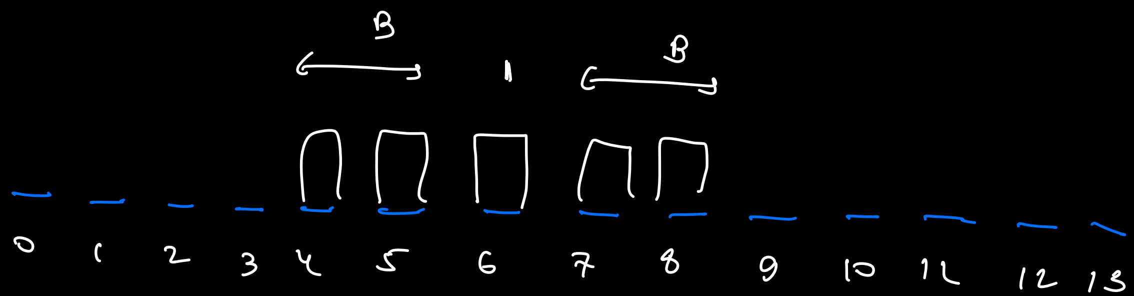
Approach: for every index, check if it can act as a centre of 2B+1 length alternating array.



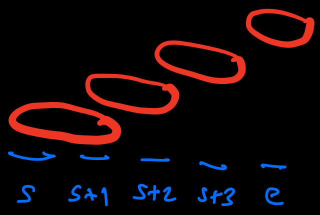
5 - B = 5 - 2 = 3

TC : O(N^2)
SC : O(N)
↙ if ans array is considered

```
int ans.

for (i=0; i<N; i++)
{
    // check if ith element can be a centre of 2B+1 len alternating subarray

    if ( i-B ≥ 0 && i+B < N)
    {
        // i is centre of 2B+1 length.
        if ( alternating( i-B, i+B, arr) == True)
        {  ans.add(i)
    }
}

return ans.
```



```
bool alternating ( int s, int e, int arr[] )
{
    for ( i = s+1 ; i ≤ e; i++)
    {
        if ( arr[i] == arr[i-1])   return False
    }

    return True.
}
```



Q. Count Increasing Triplets.

You are given an array A. Find no of triplets i, j & K s.t $i < j < K$ & $arr[i] < arr[j] < Arr[K]$.

$$A = \begin{array}{cccc} 0 & 1 & 2 & 3 \\ \left[1, \right. & 2, & 4, & \left. 3 \right] \end{array}$$

$i = 0$
$j = 1$
$k = 2$

$i = 0$
$j = 1$
$k = 3$

$i = 1$
$j = 2$
$k = 3$

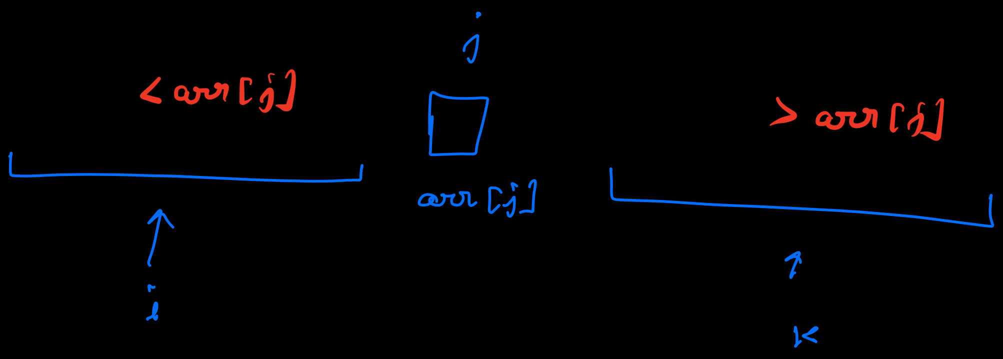
ans = 2

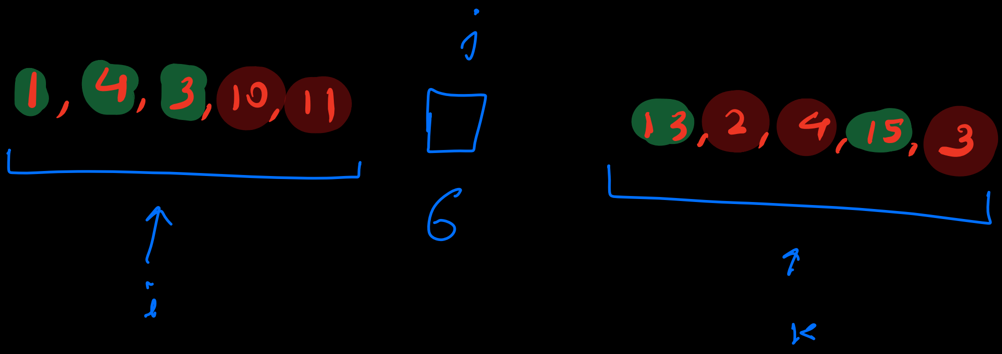
1. Consider all triplets.

TC : $O(N^3)$
SC : $O(1)$

$$(\underset{N}{10^3})^3 = 10^9$$

$\approx$ 10 seconds





1 6 13
1 6 15] 2

4 6 13
4 6 15] 2

3 6 13
3 6 15] 2

No of elements < arr[j] in left

2×3 = 6.

No of elements in right > arr[j]

```
for ( x=0 ; x<N ; x++ )
{
    // x is centre.
    left = 0.
    for( i=0 ; i<x ; i++)
    {
        if( arr[i] < arr[x]) left ++
    }
    right = 0.
    for( i=x+1; i<N ; i++)
    {
        if( arr[i] > arr[x]) right ++
    }
}
```

TC: $O(N^2)$

SC: $O(1)$

ans = ans + lft * right

return ans

Q. Array A [heights of trees]

Array B [cost of each trees]

A[i] is height of i^{th} tree & B[i] is cost of i^{th} tree.

3 trees. p, q, r

$p < q < r$ & $A_p < A_q < A_r$

Find triplet which have min cost. Return cost.

$B_p + B_q + B_r$.

1. Approach 1: consider all triplets, maintain one with min cost.

TC: $O(N^3)$.

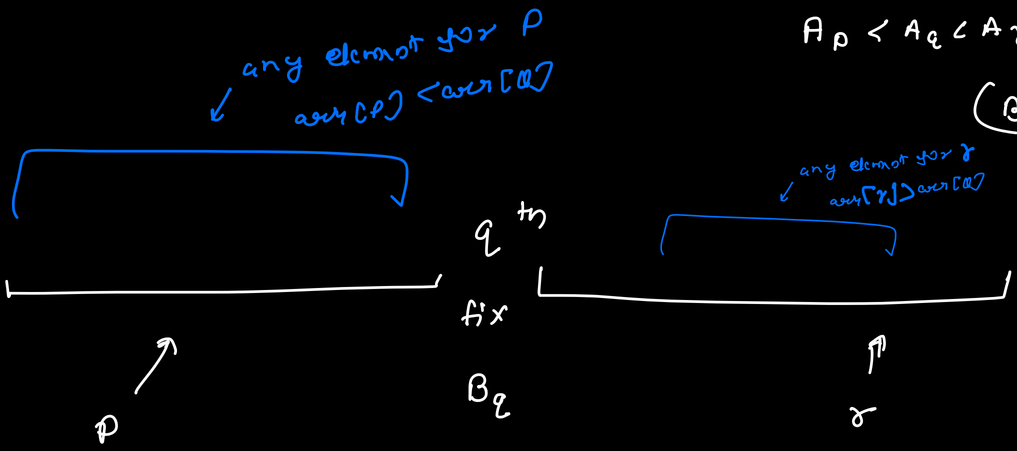
$$\underbrace{(3 \times 10^3)}_{N}{}^3 = 27 \times 10^9$$

$\approx$ 270 second

TLE

$A_p < A_q < A_r$

$(B_p + B_q + B_r)_{min}$



```
ans = INT_max
for (q=0; q<N; q++ )
{
    // q is centre.
    left = INT_max, right = INT_max
    for( p=0; p<q; p++)
    {
        if( A[p] < A[q])
        {
            left = min( left, B[p])
    for( r = q+1, r<N; r++)
    {
        if( A[r] > A[q])
        {
            right = min(right, B[r])
    tempans = left + right + B[q]
    ↑
    when q is centre.
    ans = min (ans, tempans)
}
```

TC: $O(N^2)$

SC: $O(1)$

Q. Array A of size N. Pick B elements in total. We can pick some (possibly 0) from start and some elements from last.

Find max possible sum.

A = [5, -2, 3, 1, 2]

B = 3

6 ← 0 3 [5, -2, 3, 1, 2]

8 ← 1 2 [5, -2, 3, 1, 2]

5 ← 2 1 [5, -2, 3, 1, 2]

6 ← 3 0 [5, -2, 3, 1, 2]

ans = 8.

0 1 2 3 4 5 6 7 8 9 10 N-1

A = [1, 2, 8, 7, 6, 5, 4, 3, 2, 1, -1, 8]

B = 6.

A = [1, 2, 8, 7, 6, 5, 4, 3, 2, 1, -1, 8]

A = [1, 2, 8, 7, 6, 5, 4, 3, 2, 1, -1, 8]

A = [1, 2, 8, 7, 6, 5, 4, 3, 2, 1, -1, 8]

⋮

A= [1, 2, 8, 7, 6, 5, 4, 3, 2, 1, -1, 8]

```
sum = 0
for( i= 0; i< B; i++)
{   sum= sum+ arr[i]

ans = sum

for( i=1; i≤ B; i++)
{
    sum = sum - arr[B-j]
    sum = sum + arr[N-j]
    ans= max(ans, sum)
}
return ans.
```

TC: O(B)
SC: O(1)

B-1 go	N-1 income
B-2 go	N-2 income
B-3 go	N-3 income
⋮	
B-B go	N-B income

Doubts

s, e

$$\frac{s+e}{2}$$

$2B + 2$

$B = 1$