1) Google question → $A\%P$

2) Alternating subarray [2B+1, return middle index]

3) Christmas tree.

Google

Q1. Given a no. A in the form of an array of size N, and a no. P, return $A\%P$.

Constraints

→ $1 \leq P \leq 10^9$

→ $\underbrace{1 \leq N \leq 10^5}$

$\underline{\underline{1 \leq N \leq 10}}$

ex ⇒ $A = \boxed{1 \mid 2 \mid 3 \mid 4 \mid 4}$ ⇒ 12344

$P = 4$

$A\%P = 12344 \% 4 = 0$.

A array to number x ⇒ $x\%P$

no. of digits for $x = \underline{\underline{10^5}}$.

length of array ⇒ 10^5 max

maxm value ⇒ 9 9 9 9 9 9 9 9 9 9 — ——— ——— —— ——— — —— ——— —— — 10^5 times.

ex ⇒

$$\begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 8 & 4 & 3 & 6 & 8 & 9 \end{array} \Rightarrow 3843689$$

1's

10's

100's

1000's

$$\left[(3\times10^6) + (8\times10^5) + (4\times10^4) + (3\times10^3) + (6\times100) + (8\times1\text{0}) + (9) \right] \% p =$$

$$\left[(3\times10^6) + (8\times10^5) + (4\times10^4) + (3\times10^3) + (6\times10^2) + (8\times10) + 9 \right] \% p$$

additive rule ⇒ $(a+b)\%M$

$= \left[a\%M + b\%M \right] \%M$

$$\left[(3\times10^6)\%p + (8\times10^5)\%p + (4\times10^4)\%p + (3\times10^3)\%p + (6\times10^2)\%p + (8\times10)\%p + 9\%p \right] \%p$$

$\left[(a\times b)\%M = \left[a\%M \times b\%M \right]\%M \right]$

multiplicative rule.

$$= \left[\underbrace{(3\%p \times 10^6\%p)\%p}_{0} + \underbrace{(8\%p \times 10^5\%p)\%p}_{1} + \underbrace{(4\%p \times 10^4\%p)\%p}_{2} + \underbrace{(3\%p \times 10^3\%p)\%p}_{3} + \underbrace{(6\%p \times 10^2\%p)\%p}_{4} + \underbrace{(8\%p \times 10\%p)\%p}_{5} + \underbrace{9\%p}_{6} \right] \%p$$

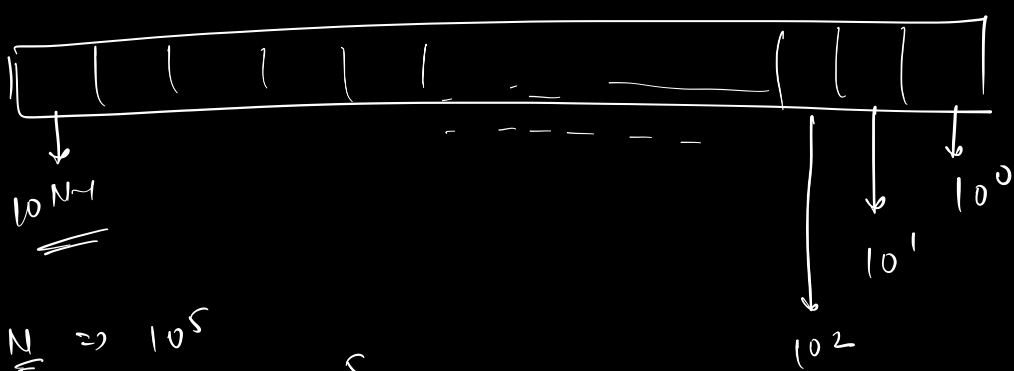
arr [3 8 4 3 6 8 9]

$(a \% p \times b \% p) \% p$

$a \downarrow$ digit or element of array

$(0-9) \% p$

$b \downarrow$ 10^k



$N \Rightarrow 10^5$

$10^k \Rightarrow 10^{10^5 - 1}$

$\Rightarrow$ func power$(a, n, p) \Rightarrow a^n \% p$

$\Rightarrow 10^k \% p$

generate array $\Rightarrow$

$$\begin{array}{cccccccccc} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & \cdots & a_{N-3} & a_{N-2} & a_{N-1} \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow \\ 10^{N-1} & 10^{N-2} & 10^{N-3} & 10^{N-4} & 10^{N-5} & 10^{N-6} & 10^{N-7} & & 10^2 & 10^1 & 10^0 \end{array}$$

index	power
0	N-1
1	N-2
2	N-3
3	N-4
4	N-5
⋮	⋮
i	N-i-1

$10^{N-i-1} \% p$

$a = 10$

$n = N-i-1$

$p = p$

$$\underbrace{(3 \% p \times 10^6 \% p) \% p}$$

$3 \to arr[i]$, $10^6 \to 10^{N-i-1}$

$\Rightarrow$ generic $\Rightarrow$ $\underbrace{(arr[i] \% p \times 10^{N-i-1} \% p) \% p}_{i^{th}\ index}$

$\downarrow$

$$\underbrace{[arr[i] \% p \times power(10, N-i-1, p)] \% p}_{i^{th}\ index}$$

$0 - (N-1)$

```
ans = 0.

for ( i=0; i<N; i++) {

    ans = ans + (arr[i] % P X power(10, N-i-1, p))%p

}
return ans % p.
```

$power(10, N-i-1, p) \to O(N)$

$O(N)$

TC $\Rightarrow O(N^2)$

$\downarrow$

Max^m $N \Rightarrow 10^5$

iteration $= (10^5)^2 = 10^{10}$ (XX)

$\downarrow$

power(a, n, p)

$\downarrow$

$a^n \% p$

power(10, N-i-1, p) $\Rightarrow 10^{N-i-1} \% p$

i	value		
0	$10^{N-1} \% p$	$\Rightarrow (10 \times 10^{N-2}) \% p$	$= (10 \% p \times 10^{N-2} \% p) \% p$
1	$10^{N-2} \% p$	$\Rightarrow (10 \times 10^{N-3}) \% p$	$= (10 \% p \times 10^{N-3} \% p) \% p$
2	$10^{N-3} \% p$	$\Rightarrow (10 \times 10^{N-4}) \% p$	$= (10 \% p \times 10^{N-4} \% p) \% p$
3	$10^{N-4} \% p$	$\Rightarrow (10 \times 10^{N-5}) \% p$	$= (10 \% p \times 10^{N-5} \% p) \% p$
4	$10^{N-5} \% p$	$\Rightarrow (10 \times 10^{N-6}) \% p$	$= \underbrace{(10 \% p \times 10^{N-6} \% p) \% p}$
⋮			

0 1 2 3 4

$\downarrow$

$10^{N-4} \% p$ $10^{N-5} \% p$

a + b + c + d $\longrightarrow$

d + c + b + a $\longrightarrow$

$(10\%p \times V_0)\%p$

i	power of 10
N-1	$10^0\%p \Rightarrow 1\%p \Rightarrow 1$
N-2	$10^1\%p = (10\times 10^0)\%p = (10\%p \times 10^0\%p)\%p$
N-3	$10^2\%p = (10\times 10^1)\%p = (10\%p \times 10^1\%p)\%p$
N-4	$10^3\%p = (10\times 10^2)\%p = (10\%p \times 10^2\%p)\%p$
N-5	$10^4\%p = (10\times 10^3)\%p = (10\%p \times 10^3\%p)\%p$

carry forward

power for that index

value (V) $\Rightarrow 10^{k}\%p \rightarrow$ ~~power func~~

```
v = 1
for (i = N-1; i >= 0; i--) {
    ans = ans + (arr[i] % p × v) % p;
    v = (10 % p × v) % p
}
return ans % p.
```

$v = 10^{N-i-1} \% p$

TC $\Rightarrow$ N-1 to 0 $\Rightarrow O(N)$

SC $\Rightarrow O(1)$

* divisibility rules
* power func
* carry forward

* Alternating Subarrays:

Given a binary array of length N, and a no. B

return all the mid indexes of alternating subarrays of length 2B+1.

ex ⇒ [0 1 0 1] ⇒ ✓

[1 0 1 0] ⇒ ✓

[0 1 0] ⇒ ✓

[1] ⇒ ✓

[0 0 1 0] ⇒ X

[1 1 0 0] ⇒ X

[1 0 1] ⇒ ✓

$$\begin{array}{c} \\ \text{ex} \Rightarrow A = \end{array} \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ {[}1 & 0 & 1 & 0 & 1{]} \\ & 1 & 2 & 3 & \end{array}$$

B = 1

length ⇒ 2B+1 = 3.

O/P = 1, 2, 3

ex ⇒ A = [0 0 0 1 1 0 1]

B = 0.

len = 2B + 1 = 1.

O/P = 0 1 2 3 4 5 6

```
ans[];
len = 2B+1

for(i=0; i<(N-len+1); i++){

    prev = -1
    flag = 1
    for(j=i; j<(i+len); j++){

        if(arr[j] == prev){
            flag=0; break.
        }
        prev = arr[j]

    }
    if(flg==1){
        ans.push(i+B)
    }
}
```

N

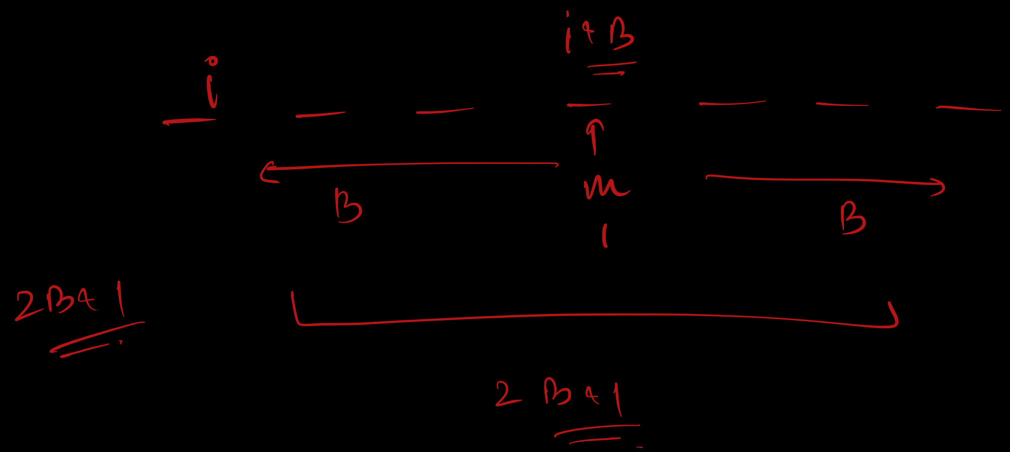
sub → k

how many subarrays

$n-k+1$

$TC \Rightarrow O(N*(2B+1))$

$SC \Rightarrow O(N)$



1 0 0

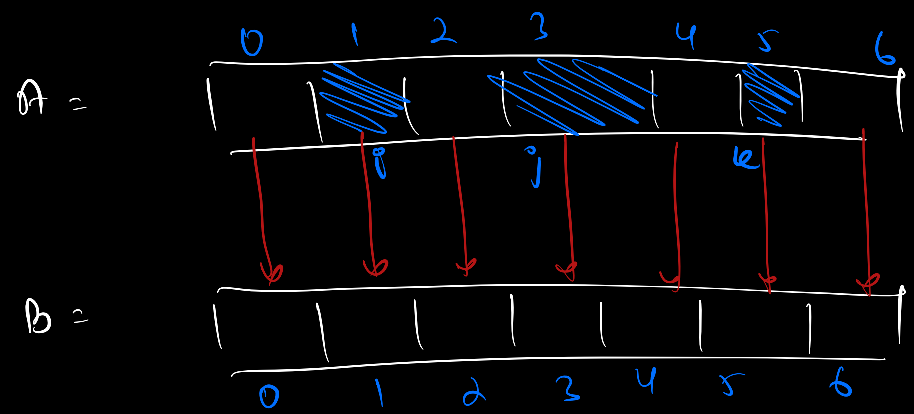
Christmas Tree

Q3. You are given an arr A, consisting heights of Christmas trees, and arr B consisting of their costs respectively, buy 3 trees such that, (i, j, k)

$(i < j < k)$

$(A[i] < A[j] < A[k])$

cost should be minimal.



Rule 1 ⇒ $i < j < k$

Rule 2 ⇒ $A[i] < A[j] < A[k]$

Rule 3 ⇒ $B[i] + B[j] + B[k]$ is minimal

ex ⇒ 1 ⇒

	0	1	2	3
A =	1	2	4	6
B =	1	9	1	2

index[i] →	{0,1,2}	{0,1,3}	{0,2,3}	{1,2,3}
values [A[i]] →	1, 2, 4	1, 2, 6	1, 4, 6	2, 4, 6
cost ⇒	1+9+1 = 11	1+9+2 = 12	1+1+2 = 4	9+1+2 = 12

ans = 0, 2, 3

ex ⇒

	0	1	2	3	4
A =	3	10	1	9	4
B =	2	5	1	3	11

$i < j < k$ $A[i] < A[j] < A[k]$

No possible triplet ⇒ o/p = -1

∘ Brute force

try all possible triplets ?

$i \rightarrow$

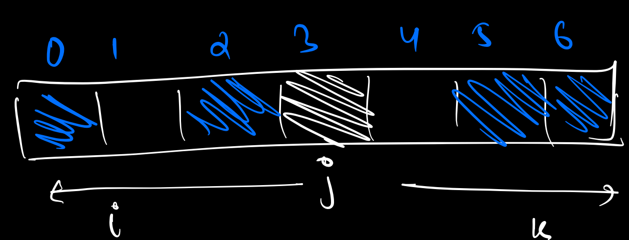
$j \rightarrow$

$k \rightarrow$

TC ⇒ $O(N^3)$

SC ⇒ $O(1)$

∘ A.



$i < j < k$

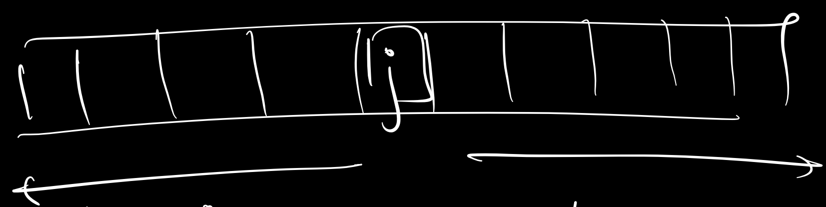
$A[i] < A[j]$ $A[j] < A[k]$.

B = take $i = 0, 2 \rightarrow \boxed{i}$

choose out of valid indexes, which has lowest cost.

$k = \underline{\underline{5, 6}}$

choose lowest value of B[k].



lowest i cost and valid

lowest k cost and valid.

ex ⇒

	0	1	2	3	4
A =	1	3	4	11	9
B =	1	3	2	6	2

$i \Rightarrow 0$ $\quad k = 2$

minm = 1 $\quad$ cost = 2

total = $\underline{\underline{6}}$

$i = 0$ $\quad k = 4$

cost = 1 $\quad$ cost 2

total = $\underline{\underline{5}}$

pseudo

```
for( j=1 → N-2) {
    for( p= j-1 → 0) {
        find valid i &
        minm cost for i
    }
```

$TC \Rightarrow O(N^2)$

$SC \Rightarrow O(1)$

for (p = j+1 → N-1) {

find valid k &

minim cost of k

}

total = B[i] + B[j] + B[k]

ans = min (ans, total)

}