Contest Discussion - 1

Question 1

```
j=0;
for(int i=0; i<n; ++i) {
    while( j<=i) {
        print(i+j)
        j++
    }
}
```

$O(n^2)$

$O(n \log n)$

$O(1)$

$O(n)$ ✓

j=0

i=0	loop	j
	1	1
1	1	2
2	1	3
	⋮	
i	1	i+1
	⋮	⋮
n-1	1	n

total iteration = O(N)

Question 2 : Range Primes

a[N], Q queries → [L, R] → count beautiful in range

Beautiful no. → prime no.

$1 <= N <= 10^4$

$1 <= A[i] <= 10^4$

$1 <= L <= R <= N$ → [1, N] not [0, n-1]

↓ -1

[0, n-1]

Code

```
int prime[n]

for(int i=0; i<n; ++i) {          → O(n * sqrt(a[i]))
                                  O(n * sqrt(10^4))
                                  ↓
                                  O(n sqrt(n))
    // a[i] is prime or not?
    prime[i] = 1  // initially, assume a[i] is prime
    for(j=2; j*j <= a[i]; ++j) {
        if(a[i] % j == 0) {
            prime[i] = 0;
            break;
        }
    }
    if(a[i] == 1) {
        prime[i] = 0;
    }
}
```

a[] = 5 9 11 20

prime[] = 1 0 1 0

```
for(i=1; i<n; ++i) {                    → O(n)
    prime[i] += prime[i-1]              → calculate prefix sum array of prime array
}
int ans[n]
for(i=0; i<q; ++i) {                    → O(q)
    L → B[i][0]  , R → B[i][1]
    l = B[i][0] , r = B[i][1]
    l--1, r--1   → since l,r are 1-indexed, reduce 1 to make 0-indexed
    if(l==0) {
        count = prime[r]
    }
    else {
        count = prime[r] - prime[l-1]
    }
    ans[i] = count
}
return ans
```

TC: $n\sqrt{n} + n + q$

: $O(n\sqrt{n} + q)$

$n = 10^4$
$q = 10^4$

$10^4\sqrt{10^4} + 10^4 = 10^6 + 10^4 \approx 10^6$

SC: $O(n)$

Question3 : Longest Magical Subarray

a[n] → max size subarray with only magical no.

magical → divisible by 2 or 3.

$1 <= N <= 10^5$

$1 <= A(i) <= 10^9$

Code

```
int magical[n]
for(i=0; i<n; ++i) {                    → O(n)
    if(a[i]%2==0 || a[i]%3==0) {
        magical[i] = 1
    } else {
        magical[i]=0
    }
}
ans=0, len=0
for(i=0; i<n; ++i) {                    → O(n)
    if(magical[i]==1) {
        ++len;
    } else {
        len=0
    }
```

→ not needed (optional)

repeat

in case of this

ans = max(ans, len)

}

return ans

TC : O(N)

SC : O(N) → could be O(1)

ans=0, len=0

magical(1) :