Optional Problem Solving Session

1) Recordings will be available

2) Problems to be discussed.

17% - a) Delete one ✓ 1

20% - b) Very Large power ✓ 2

22% - c) All GCD pair ✓ 3

21% - d) Max Distance → Hint1 / Hint2 / (understand)

20% - e) Merge Intervals.

Doubt Session

i) Detailed idea /

10:35 break

3) Note: Assignment problems on primes & permutations please try for some more time, since they were only discussed last week.

a) Delete one:

Given N array elements, we have to delete 1 element such that GCD (Greatest Common Divisor) of remaining array is max.

Ex: $ar[5]$:

0	1	2	3	4
24	16	18	30	15

→ ans = 3

24 ✗, $gcd(\underbrace{16\ 18\ 30\ 15}) \rightarrow 1$

$ar[5]$: 24 16 18 30 15 → $g(\underbrace{24})$, 16 ✗, $g(\underbrace{18\ 30\ 15}) \rightarrow 3$

$ar[5]$: 24 16 18 30 15 → $g(\underbrace{24\ 16})$, 18 ✗, $g(\underbrace{30\ 15}) \rightarrow 1$

$ar[5]$: 24 16 18 30 15 → $g(\underbrace{24\ 16\ 18})$, 30 ✗, $g(\underbrace{15}) \rightarrow 1$

$ar[5]$: 24 16 18 30 15 → $g(\underbrace{24\ 16\ 18\ 30})$, 15 ✗ } gcd = 2

Idea : 1) Remove an element & gcd of entire array.

$N * [\text{gcd of Entire array}]$ → max of array.

→ $O(N * \log_2(\text{Max}))$

Instead of Removing an element while taking gcd of Entire array neglect the element we want to delete

// Assume N = 7 : 0 1 2 3 4 5 6

Delete	gcd
	gcd of all Elements
0	gcd[1-6] : from index 1-6
1	gcd(gcd[0-0], gcd[2-6])
2	gcd(gcd[0-1], gcd[3-6])
3	gcd(gcd[0-2], gcd[4-6])
4	gcd(gca[0-3], gcd[5-6])
5	gcd(gcd[0-4], gcd[6-6])
6	gcd[0-5]

out of all Then gcd's get max.

// prefix : starts at 0^{th} index.

// Hint 1 : pregcd[i] = gcd of all elements from index [0 - i]

TC: $O(N \cdot \log_2 Max)$

pregcd[0] = ar[0]

i = 1 ; i < N ; i++) {

pregcd[i] = gcd (ar[i], pregcd[0, i-1])

}

// suffix : ends at $N-1^{th}$ index.

Hint 2 : sufgcd[i] = gcd of all elements from index [i, N-1]

TC : $O(N \cdot \log_2 Max)$

sufgcd[N-1] = ar[N-1]

i = N-2 ; i >= 0 ; i--) {

$\underbrace{\text{sufgcd[i]}}_{\text{gcd (i, N-1)}}$ = gcd (ar[i] , $\underbrace{\text{sufgcd[i+1]}}_{\text{gcd (i+1, N-1)}}$)

ar[i] gcd (i+1, N-1)

}

// Hint 3 : For every element calculate gcd of remaining array & get max.

O(N)

Overall TC: O(N + N + N) → O(N) SC: O(N)

// Rough Notes

gcd of entire array = $(N * \log_2 max)$

```
ans = ar[0]
i=0; i<N; i++) {
    ans = gcd(ans, ar[i])
}
return ans;
```

	itr
gcd(4, 4) = 4	1
gcd(8, 8) = 8	1
gcd(8, 40) = 8	1
gcd(2, 50) = 2	1

ar[1] | ar[2] . . . | ar[N-1]

1) ans = ar[0]

2) ans = 1

3) ans = gcd( )

gcd(a, b) = gcd(a, b%a)

Note1: Give a, p if p is prime & $gcd(a,p)=1$

Inverse Modules : $\{ (a^{-1}) \% p = \underbrace{(a^{p-2}) \% p}_{} \}$ { Doubts Session }

↳ use fast power exponentiation

Note2: Give a, p if p is prime & $gcd(a,p)==1$

p is not a factor of a

$\{ (a^{p-1}) \% p = 1 \}$

Both above are based fermat's little theorm

$(a^{-1}) \% p = (a^{p-2}) \% p$

Multiply a with both

$1 = (a^{p-1}) \% p$

208) **Very Large power:**

Given 2 Integers A, B you have to calculate

$\{A, B < = 5*10^5\}$ $\quad (A^{B!}) \% (10^9+7)$

$A < = 5*10^5 < P\}$ $\quad \gcd(A, P) = 1$ $\quad P$ → prime

1) $(A^{P-1}) \% p = 1$

2) $(A^{P-1})^2 \% p$? → $((A^{P-1}) \% p * (A^{P-1}) \% p) \% p = 1$

3) $(A^{P-1})^3 \% p$? = 1

Q) $(A^{B!}) \% p$? $\quad$ $\boxed{\text{know} \rightarrow (A^{P-1}) \% p = 1}$

$\boxed{r = B! \% (P-1)}$

B! = divident

P-1 = divisor

$\boxed{B! = (P-1) * q + r}$

$(A^{(P-1)q + r}) \% p$

$A^{m+n} \rightarrow a^m * a^n$

$$\left(A^{(p-1)q+r}\right) \% p$$

$$= \left(\underbrace{\left(A^{(p-1)q}\right) \% p}_{} \times \left(A^{r}\right) \% p\right) \% p$$

$$= \left(\underbrace{\left(\left(A^{p-1}\right)^{q}\right) \% p}_{1} \times \left(A^{r}\right) \% p\right) \% p$$

$$= \left(\underbrace{1}_{} * \left(A^{r}\right) \% p\right) \% p$$

$$= \left(A^{r}\right) \% p \qquad \underbrace{r = (B!) \% (p-1)}_{}$$

↳ iterate & get value of r.

$$= \left(A^{r}\right) \% p \; \Big\} \rightarrow$$ using fast power exponentiation

TC: 1) Calculate $r = (B)! \% p-1 \Rightarrow TC: O(B)$

2) Calculate $\left(A^{r}\right) \% p \Rightarrow TC: O(\log B)$

$TC \Rightarrow (B + \log B) \qquad SC: O(1)$

$\underline{2^{nd}}$ approach :

$$a^{3!} = a^{1*2*3} = ((a^1)^2)^3$$

$$a^{4!} = a^{1*2*3*4} = (((a^1)^2)^3)^4$$

$$a^{B!} = a^{1*2*3*\cdots B} = \Big\{\underbrace{((((a^1)^2)^3)^4)^{\cdot^{\cdot^{N}}}}\Big\}$$

→ power (a, b, p) : It calculate $a^b \% p$ In $\log b$

ans = a; $p = 10^9+7$

$b = 4$

$i = 1$ ans = a^1

$i = 2$ ans = a^2

$i = 3$ ans = $(a^2)^3 = a^6$

$i = 4$ ans = $(ans)^4 = (a^6)^4$
$= a^{24}$

```
i = 1; i <= b; i++) {        (1—B)
    ans = power(ans, i, p)
}
```

↳ ans = $(a^{B!}) \% p$

Max $B = 5*10^5$

TC : $B * \log_2(B)$

SC : $O(1)$

$$5*10^5 * \log_2(5*10^5)$$

$$5*10^5 * \{\log_2 10^5 + \log_2 5\}$$

$$5*10^5 * \{\approx 20\}$$

→ $100 * 10^5 = \boxed{10^7 \text{ iterations} < 10^8}$

All GCD Pair:

Given N elements which contains gcd of all pairs of another array, find initial array using which gcd is calculated

ar[4] = { 12, 3, 6, 8 }

Input $x[\ ]$ {

gcd(12, 12) gcd(12, 3) gcd(12, 6) gcd(12, 8)
12 3 6 2

gcd(3, 12) gcd(3, 3) gcd(3, 6) gcd(3, 8)

gcd(6, 12) gcd(6, 3) gcd(6, 6) gcd(6, 8)

gcd(8, 12) gcd(8, 3) gcd(8, 6) gcd(8, 8) }

$x[\ ] = \{ 12, 3, 6, \cancel{8}\ 4, 3, 3, 3, 1, 6, 3, 6, 2, 4, 1, 2, 8 \}$

All gcd pairs

$x[\ ] = \{ 8, 2, 1, 3, 3, 6, 6, 12, 3, 3, 2, 3, \cancel{8}\ 4, 6, 1, 4 \}$

→ Not in order

→ find Initial array Elements

16 example } N = 4

$\{\{ a_1 >= a_2 >= a_3 >= a_4 \}\}$ Decreasing order

$n[\,] = \{$ [illegible] $\}$

$ar[\,] = \{\{12, 8, 6, 3\}\}$

	a_1	a_2	a_3	a_4
a_1	$gcd(a_1, a_1)$ 12	$\cancel{gcd(a_1, a_2)}$ $\cancel{\alpha = a_2}$	$\cancel{gcd(a_1, a_3)}$ $\cancel{\alpha = a_3}$	$gcd(a_1, a_4)$ $\alpha = a_4$
a_2	$\cancel{gcd(a_1, a_2)}$ $\cancel{\alpha = a_2}$	$gcd(a_2, a_2)$ $= a_2$	$\cancel{gcd(a_2, a_3)}$ $\cancel{\alpha = a_3}$	$gcd(a_2, a_4)$ $\alpha = a_4$
a_3	$\cancel{gcd(a_3, a_1)}$ $\cancel{\alpha = a_3}$	$\cancel{gcd(a_3, a_2)}$ $\cancel{\alpha = a_3}$	$\boxed{gcd(a_3, a_3)}$ $= a_3$	$gcd(a_3, a_3)$ $\alpha = a_4$
a_4	$gcd(a_1, a_4)$ $\alpha = a_4$	$gcd(a_2, a_4)$ $\alpha = a_4$	$gcd(a_3, a_4)$ $\alpha = a_4$	$gcd(a_4, a_4)$ $= a_4$

// n[] → This contains all gcd pairs ( )

arr[] → ______, N = sqrt(n.size())

while(arr.size() != N) {

iterate & get max in n[] = ele } 1st Element in your n[]

delete gcd(ele, ele) in n[]

i = 0; i < arr.size(); i++) {

int y = gcd(arr[i], ele)

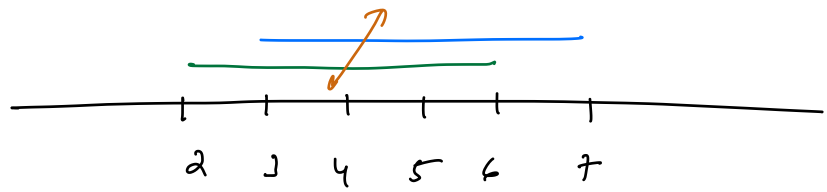
// delete 2 occurrences of y in n[]

}

arr.insert(ele)

}

// Intervals :



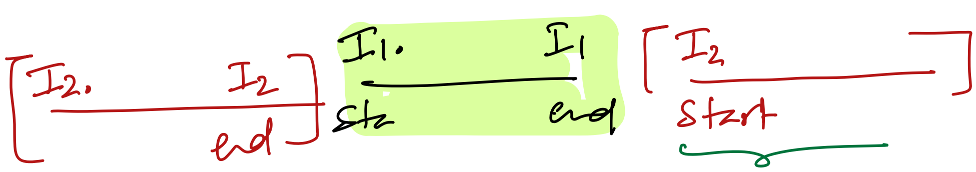
I_1	I_2	Final Interval
(2, 6)	(3, 7)	(2, 7)
(2, 8)	(4, 6)	(2, 8)
(3, 7)	(4, 10)	(3, 10)
(3, 6)	(6, 10)	(3, 10)

(2, 5)   (8, 10)   * Not overlapping :

I_2.start > I_1.end } no overlapping

(5, 8)   (1, 3)   * Not overlapping :

I_2.end < I_1.start } no overlap.



I_2.end < I_1.start   or I_2.start > I_1.end

If not above 2 case, they have to overlap.

Merge Intervals

Given N non overlapping intervals & insert a new interval

→ Intervals are sorted based on start

Eg: N = 7 new Interval

✓ [1 3] < [12 22]

✓ [4 7] < [12 22]

[10 14] — [12 22]

merged

[16 19] — [10, 22]

merged

[21 24] — [10, 22]

merged

[27 30] [10, 24]

[32 35]

Ans:

[1 3]

[4 7]

[10, 24]

[27 30]

[32 35]

Ex2: N = 8 newInterval

[1 5] < [23 36]

comes first

[8 10] < [23 36]

[11 14] < [23 36]

[15 20] < [23 36]

[22 24] [23 36]

merge

[26 28] [22, 36]

merge

[29 34] [22, 36]

merge

[35 37] [22, 36]

merge

[22, 37]

ans.

[1 5]

[8 10]

[11 14]

[15 20]

[22 37]

// pseudocode

// say given N intervals ar[ ] , Input newInterval

ar[i].start , ar[i].end

newInterval.start

newInterval.end

We need ans< > ⇒ TC: O(N) SC: O(N)

SC: O(1)

```
i = 0; i < N; i++) {
   // When can we say ar[i] Interval comes before new interval
   if (ar[i].end < newInterval.start) {
        ans.insert(ar[i])
   }
   When can we say ar[i] Interval comes after new interval
   // else if ( newInterval.end < ar[i].start) {
        ans.insert( new Interval)
        j = i; j < N; j++) {
            ans.insert(ar[j])
        }
        return ans;
   }
   // overlapping case
   else {
        newInterval.start = min(newInterval.start, ar[i].start
        newInterval.end = max(newInterval.end, ar[i].end)
   }
}
ans.insert(newInterval); return ans;
```

// All gcd pairs ( )

$i = \min(a_1, a_2)$; $i >= 1$; i--

if ($a_2 \% i == 0$

&& $a_1 \% i == 0$)

1) $\gcd(a, a) = a$

2) $\gcd(a_1, a_2)$ $\alpha = \min(a_1, a_2)$

3 points

3) $a_1 >= a_2$

$\gcd(a_1, a_2)$ $\alpha = a_2$

4) $a_1 >= a_2 >= a_3$

1st max

$\gcd(a_1, a_1)$ | $\cancel{\gcd(a_1, a_2)}$ $\cancel{\alpha = a_2}$ | $\gcd(a_1, a_3)$ $\alpha <= a_3$

$\cancel{\gcd(a_2, a_1)}$ $\cancel{\alpha = a_2}$ | $\gcd(a_2, a_2) = a_2$ | $\gcd(a_2, a_3)$ $\alpha \geq a_3$

$\gcd(a_3, a_1)$ $\alpha = a_3$ | $\gcd(a_3, a_2)$ $\alpha = a_3$ | $\gcd(a_3, a_3) = a_3$

2nd max