No. of pairs

Inversion count

No. of open doors

↓

Prime Nos. Assignment

Contest → Next Fri 3 Nov

9 - 10:30 PM

↓

Syllabus - Recursion, Maths

Q. Given two array A[n] and B[m]. Calculate no. of pairs such that A[i] > B[j]

A[3] = < 7 3 5 >

B[3] = < 2 0 6 >

larger in A

↓

smaller in B

A	B		
7, 2	3, 2	5, 2	
7, 0	3, 0	5, 0	
7, 6			

ans = 7 pairs

A = 3 6 8 10 15 L

B = 1 2 7 12 18 S

3, 1	6, 1	8, 1	10, 1	15, 1
3, 2	6, 2	8, 2	10, 2	15, 2
		8, 7	10, 7	15, 7
				15, 12

ans = 14 pairs

BF : Consider all pairs (2 loops)

i loop → fix an element in A

j loop → fix an element in B

```
if (A[i] > B[j])
    cnt ++
```

TC : O(NM)

SC : O(1)

$1 \le N, M \le 10^5$

A = 8 3 6 10 15

B = 1 2 7 12 18

0 1 3 4 6 8 10 Sorted

2 larger elements

sort A and B

N = 5 A : 3 6 8 10 15

M = 5 B : 1 2 7 12 18

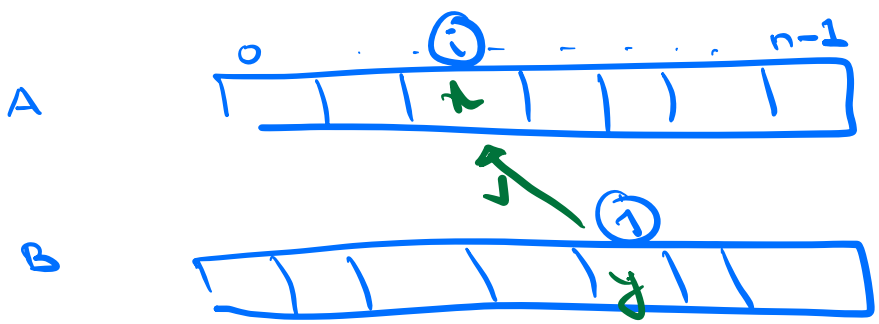
ans = ~~0~~ ~~0~~ ~~10~~ ~~13~~ 14

	0	1	2	3	4	5	6	7	8	9
C	1	2	3	6	7	8	10	12	15	18
	+5	+5			+3			+1		

B B A A B A A B A

[a b]

b-a+1



elements from $i \to n-1$

$n - \cancel{x} - i + \cancel{x}$

$= n - i$

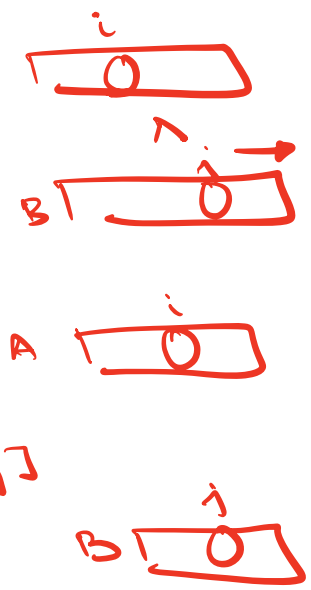
```
sort (A)
sort (B)
merge (A, B, N, M)

int [] merge (int A[], int B[], int N, int M) {
    int ans = 0
    int C[N+M]
    int i = 0, j = 0, k = 0

    while (i < N && j < M) {
        if (A[i] ≤ B[j]) {
            C[k] = A[i]
            k++    i++
        }
        else {      // A[i] > B[j]
            C[k] = B[j]
            k++    j++
            ans += N - i
        }
    }

    // loop will end if i reaches
    end of A or j reaches end of B

    while (i < N) {  // check if A is remaining
        C[k] = A[i]
        k++    i++
    }
```



```
while (j < M) {   // check if B is remaining
    C[k] = B[j]
    k++    j++
}
}
```

10:46

TC : $O(N \log N + M \log M + N + M)$

SC : $O(N + M) \rightarrow O(1)$

in this code — without array C

Count logic

```
while (i < N && j < M) {
    if (A[i] ≤ B[j]) {
        C[k] = A[i]
        k++    i++
    }
    else {   // A[i] > B[j]
        C[k] = B[j]
        k++    j++
        ans += N - i
    }
}
```

Q. Given an A[n], calculate no. of pairs i, j such that $i < j$ and $a[i] > a[j]$

$a[5] = \langle 10 \quad 3 \quad 8 \quad 15 \quad 6 \rangle$

+3 +0 +1 +1 +0

indices

i & j

ans = 5

i j

$A[i] > A[j]$

i j

Left idx Right idx

Larger > Smaller

BF : Iterate all (i, j) pairs → 2 loops

$i < j$

```
for (i=0 ; i<n ; i++) {
    for (j= i+1 ; j<n ; j++) {
        if (A[i] > A[j])
            cnt++
    }
}
```

TC : $O(N^2)$

SC : $O(1)$

a[5] = 10 3 8 15 6

+1

3 10 8 15 6

+1

3 8 10 15 6

3 8 10 15 6

+1

+1

+1

ans = 5

3 6 8 10 15

	0	1	2	3	4
a[5] =	10	3	8	15	6

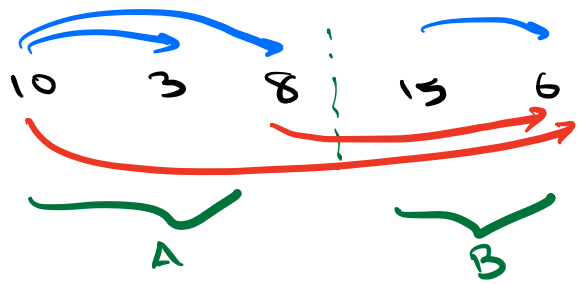
idx, val

(0,10) (1,3) (2,8) (3,15) (4,6)

sort

1, 3 4, 6 2, 8 0, 10 3, 15

j



Inversion count

$i < j$

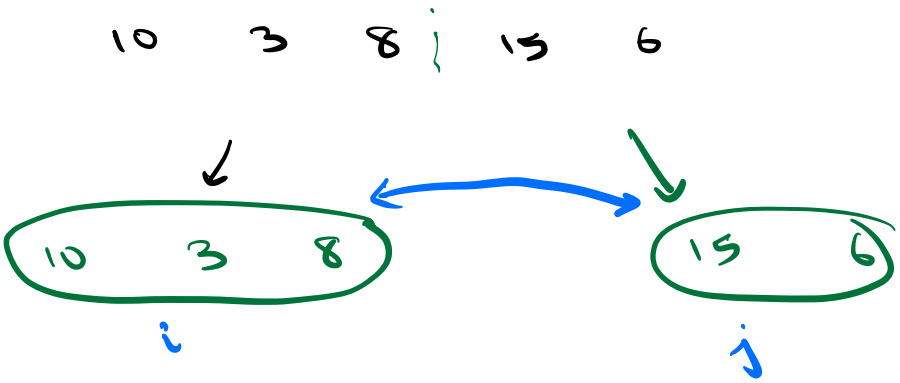
$A[i] > A[j]$

ans = $T(A) + T(B) + T(A \& B)$

$2 + 1 + 2$

$= 5$

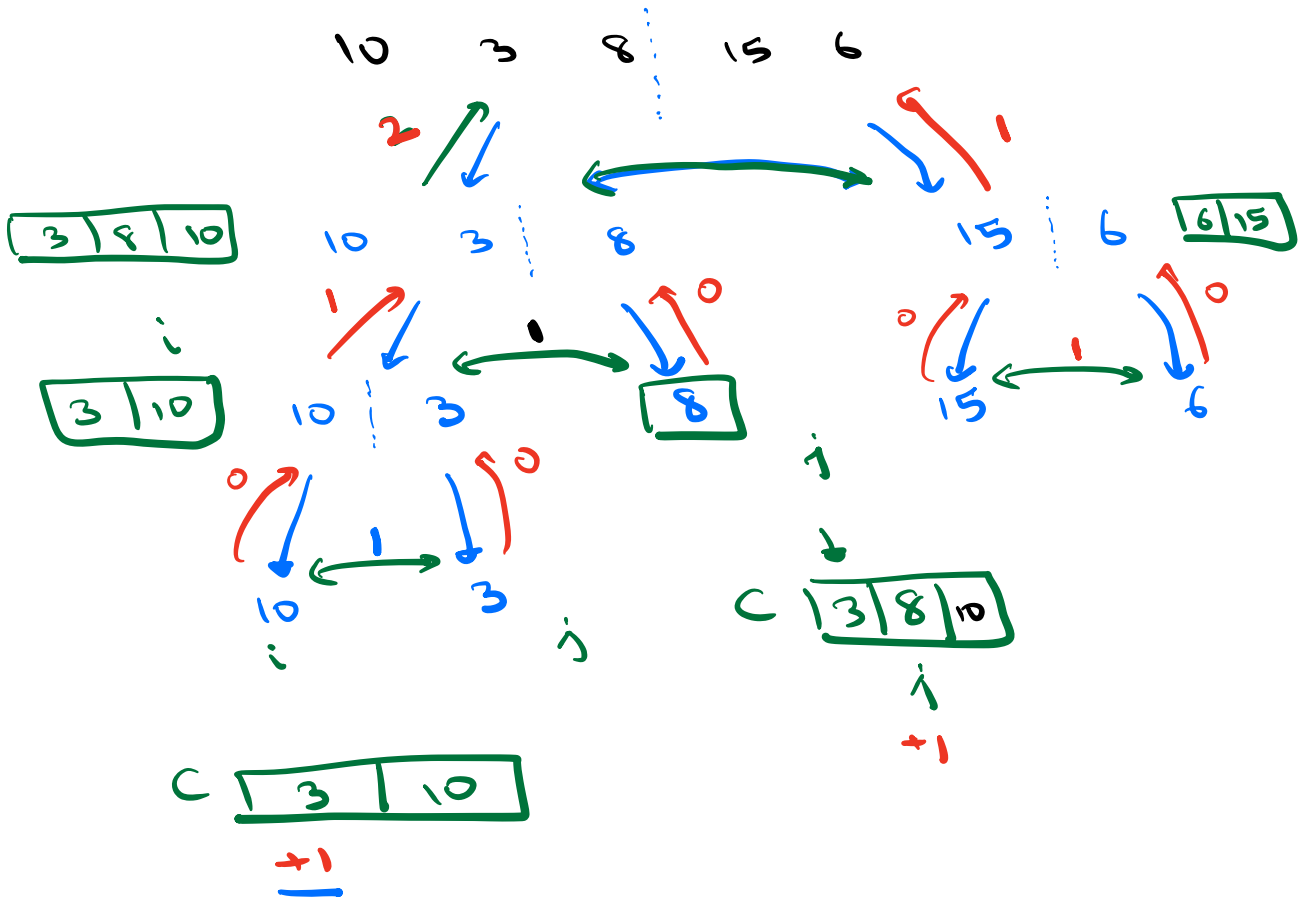
$A[i] > B[j]$

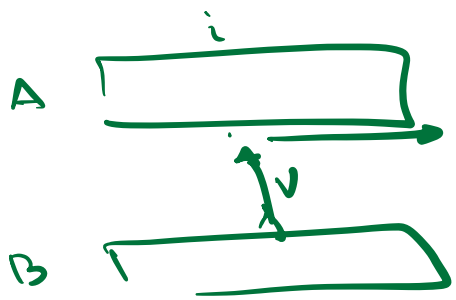


$i < j$

$A[i] > A[j]$

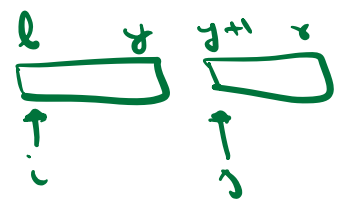
$T(A \& B) \rightarrow$ sort A & sort B

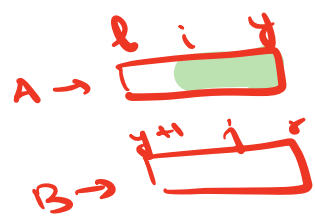

10 3 8 15 6
2
1
3 8 10
10 3 8 15 6
6 15
1
0
0
i
3 10
10 3 8 15 6
0
0
1
1
0
10 3
i j
j
C 3 8 10
i
+1
C 3 10
+1


i
A
v
B

```
// return invcount l->r , sort l->r
int invcount ( int ar[], int l, int r) {
    if (l == r)    return 0
    int mid = (l+r)/2
    int x = invcount (ar, l, mid)
    int y = invcount (ar, mid+1, r)
    int z = merge (ar, l, mid, r)
    return x+y+z
}
```

```
int merge (int A[], int l, int y, int r ) {
    int cnt = 0
    int C[r-l+1]
    int i = l, j = y+1, k = 0
    while (i <= y && j <= r) {
        if (A[i] ≤ A[j]) {
            C[k] = A[i]
            k++      i++
        }
        else {      // A[i] > B[j]
            C[k] = A[j]      cnt += y-i+1
            k++      j++
        }
    }
```





// loop will end if i reaches end of A or j reaches end of B

```
while (i ≤ y) {   // check if A is remaining
    C[k] = A[i]
    k++    i++
}

while (j ≤ r) {   // check if B is remaining
    C[k] = A[j]
    k++    j++
}
```

10:46

// copy C arr

```
int i = l , k = 0

while (i ≤ r){
    A[i] = C[k]
    i++    k++
}
```

A[l] ← C[0]

}

N
↓
N/2
↓
N/4
↓
⋮
1

TC: O(N log N)

SC: O(log N + N)

3. No. of (Prime) open doors

All doors → closed

$N = 10$

Open →
Close → No color

1	2	3	4	5	6	7	8	9	10
1	1,2	1,3	1,2,4	1,5	1,2,3,6	1,7	1,2,4,8	1,3,9	1,2,5,10

Open doors → 1, 4, 9 — perfect squares

1 to 10

Door → factors

Initially	1	2	3	4
Close	Open	Close	Open	Close

$1 \to N$ → doors with odd factors → open

24

i	N/i
1 →	24/1
2 →	12
3 →	8
4 →	6

$\boxed{N \quad i \to 1 \text{ to } \sqrt{N}}$

$i = N/i$

$\boxed{i^2 = N}$

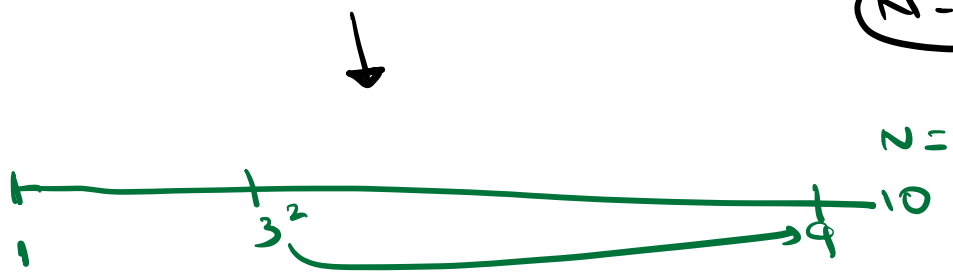
cnt += 1

100

i	N/i
10 →	10

Perfect squares in $1 \to N$

$N = 10^9$



ans = sqrt (N)

3^2 ✓ < 10 ✓

2^2 ✓

1^2 ✓

Doubt

A → 2 , 3 , 2 , 3 (indices 0, 1, 2, 3)

B → [2, 6, 2, 6] (indices 0, 1, 2, 3) → Total no. of non empty subset

subset → <2,6,6>

2 2,3 2,3

same set of prime factors

<6,6> <2,2>

<2> <6> <2> <6>