Problem solving 4

① Majority Element

Given an array of N elements.
Find majority element.

count (element) > N/2

3 3 4 2 4 4 2 4 4 → 9/2 ⇒ 4

5

1 2 3 3 4 5 → 6/2 = 3

↳ no majority element
> N/2

N/2 + 1

N/2 − 1

one single majority element can be present

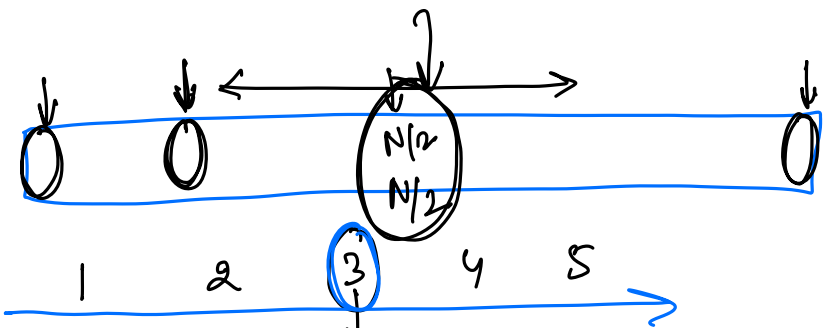
idea: ①

Hashmap < element, count >

if there is any element with count > N/2

→ T.C : O(N)
S.C : O(N)

② sorting



N/2 + 1

T.C : O(n log n)
S.C : O(1)

$O(N)$

T.C : $O(N)$
S.C : $O(1)$

RR
RR
RRR

RR

RRR

R

1) NMP ✗ NMP

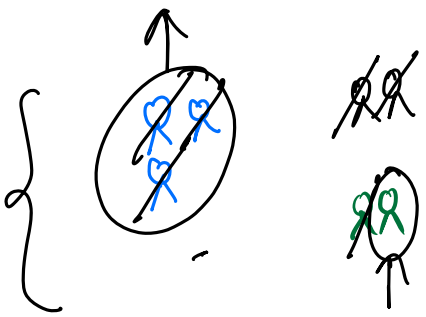
2) ME ✗ NMP

N	$N/2+1$
$N-2$	$N/2+1$

	ME
N	$N/2+1$
$N-2$	$N/2$

$\frac{(N-2)+1}{2}$

$= \frac{N}{2} \cancel{-1+1}$



	3	3	4	2	4	4	2	4	4
M.E	3	3	3	3	4	4	4	4	4
count	1	2	1	0	1	2	1	2	3

candidate sol^n

Moore's voting Algorithm

T.C : $O(N)$
S.C : $O(1)$

```
M.E = arr[0]
count = 1
for( int i=1; i<n; i++)
{
    if( arr[i] == M.E)
        count++;
    else if{ if( count==0)
             { ME = arr[i]; count = 1;}
             else
```

N

```
}  else
        cont--;
}
c=0;
for( i=0 → n)
    if( arr[i] == me)
        c++;
}
if( c > n/2)
    return me
return -1;
```

N

$2N \sim O(N)$

	1	2	3
me	1	1	3
	1	0	1

Q2 You are given 3 prime no's

$p1 < p2 < p3$

Give first N elements of the form

$p1^i \cdot p2^j \cdot p3^k$ $\quad i, j, k >= 0$

$p1 = 2$
$p2 = 3$
$p3 = 5$

$1 = 2^0 * 3^0 * 5^0$

$2^2 \times 3^0 \times 5^0$

1 2 3 4 5 6 8 9 10 12 ~~14~~ 15

$2^2\ 3^0\ 5^0$ $\quad 2*7$ $\quad 15 \to 3*5$

$2^1 * 3^1 * 5^1$ $\quad 30 \to 2 \times 3 \times 5$

2×15
$2 \times 3 \times 5$

$2^3\ 3^2\ 5 \to$

N = 7

$p1^i * p2^j * p3^k$

p1, p2, p3 prime no

n

BF

for(int i=1 → ∞)

{

i

~~if(i%p1 == 0 || i%p2 == 0 || i%p3 == 0)~~

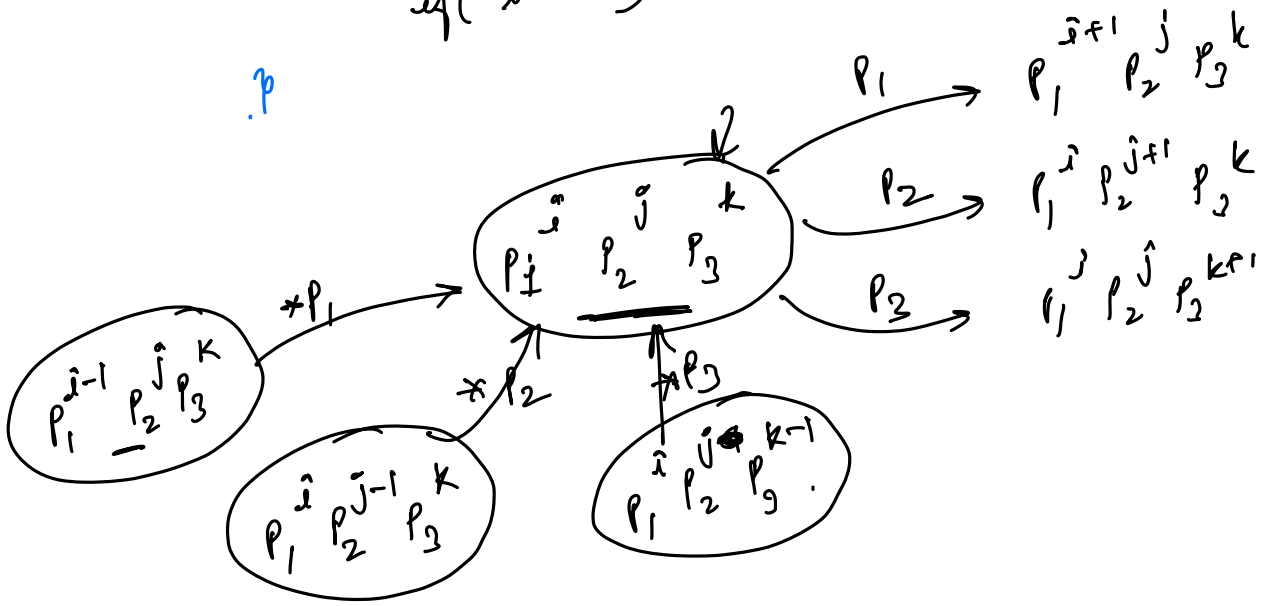
14

14/2 = 7 %3 = 0

24
12
6
3
1

while (divide by 2)
divide by 3
divide by 5
if(i == 1)

p



2,3,5

[1]
p1 p2 p3

1 * p1 = 2
1 * p2 = 3
1 * p3 = 5

2 3 5 4 6 10 9 15

min

2 p1 = 4
p2 = 6
p3 = 10

3 p1 = 6
p2 = 9
p3 = 15

2 * 3

$2^2 * 3$

12

O(n)

smart

[1]

1 * p1 = 2
1 * p2 = 3
1 * p3 = 5

[1 2]

2 * p1 = 4
1 * p2 = 3
1 * p3 = 5

[1 2 3]

p2 p3 p1 p2

2 * p1 = 4
2 * p2 = 6
1 * p3 = 5

[1 2 3 4]

p3 p2 p1

~~1 * p1 = 2~~
1 * p2 = 3
1 * p3 = 5

2 * 2
* 3
* 5

[1]

p1 p2 p3

1 * p2 = 3
1 * p3 = 5
2 * p1 = 4

p2 * 1
p2 * 2

[1 2]

p2 p3 p1

1 * p3 = 5
2 * p2 = 6
2 * p1 = 4

[1 2 3]

p3 p2 p1

3 * n

[1 2 3 4 5]

p1 p2 p1

p2 * 1 = 5
p2 * 2 = 6
p2 * 3 = 6

ans = [ ] ← 1

3 pointer

i1 = 0
i2 = 0
i3 = 0 cont = 1;

while (cont < n)
{
x = min{ p1 * ans[i1], 1
6 p2 * ans[i2], 1
6 p3 * ans[i3], } 1

if (x == p1 * ans[i1])
i1++;
else if (x == p2 * ans[i2])
i2++;
else
i3++;

ans[ ] ← x cont++;
}

last inserted element < current

if (ans[cont-1] < x)

6 6 5

Q3 3 sorted Arrays A, B, C,
choose one element from each array
such that
max(a, b, c) − min(a, b, c) ↓ minimized

A:	1	4	5	8	10
B:	6	9	15		
C:	2	3	6	8	

1, 6, 2
max = 6
min = 1
diff = 5

B.F

consider all possible tripples

p q r

```
for( int i=0 -> p)
{
    for( int j=0 -> q)
    {
        for( int k=0 -> r)
        {
            A[i] , B[j], C[k]
        }
    }
}
```

T.C: $O(n^3)$

idea ⇒ max( ) − min( )

decrease ↓ ↑ increase

minimize

5

A: 1 4 5 8 10

B: 6 9 15

C: 2 3 6 8

1 ↓ min

6 ↓ max

2

5

4 6 2

max min

{1 _ _ 6 2}

{ _ _ 2}

4 6 2

4 9 2

5 9 2

4 6 3

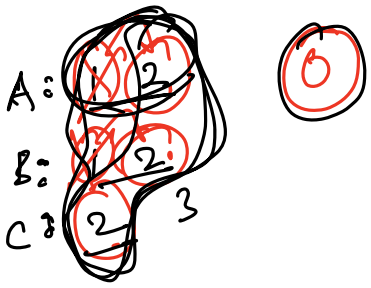
max min

4 6 6

min max

always try to minimize to min

5 4 3 2 1 → 0



0

max

T.C?

$O(p+q+r)$

$O(3N)$

$\approx O(N)$

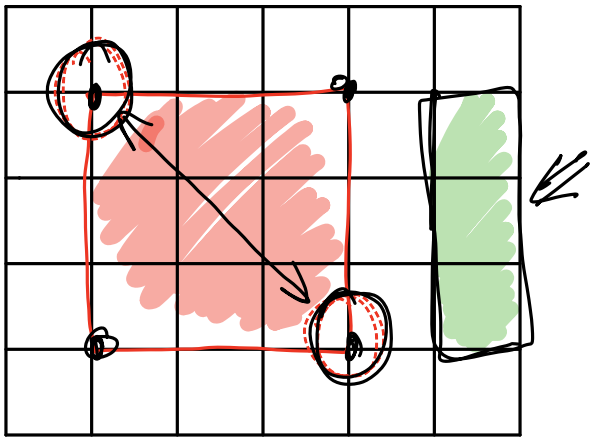
```
i=0   j=0   k=0
while( i<p && j<q && k<r )
{
    max = max( A[i], B[j], C[k] )
    min = min( ------------ )
    if( max-min < diff)
        diff = max-min
    if( diff==0)
        break;
    if( min==A[i]
        i++
    else if( min==B[j]
        j++
    else
        k++;
}
```

(4)

matrix

submatrix → two coordinate

TL & BR



q queries → two coordinates

↓

represent a submatrix

↓

total sum

q time

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

✓ (0,0) (1,2) 22

✓ (0,3) → (3,4) 36

a b c d

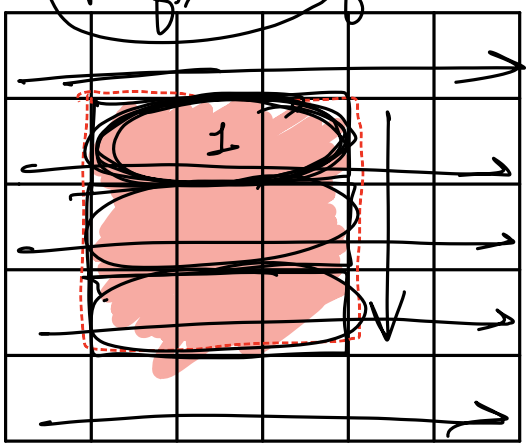
T.C: $O(q * n^2)$

ideal

iterate

```
for( i = a; i <= c; i++)
{   for ( j = b; j <= d; j++)
    {
        ___
    }
}
```

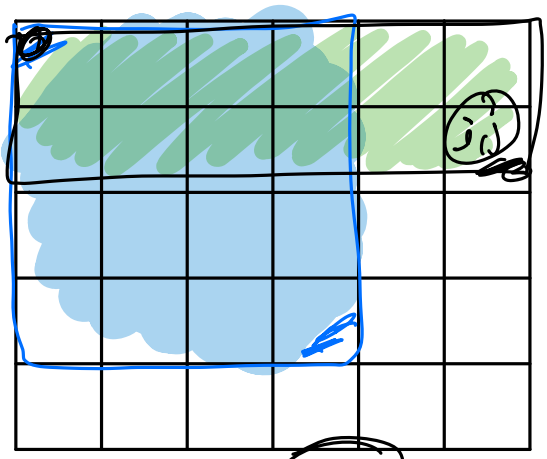
Prefix sum of every row prefix sum



$T(q * n)$

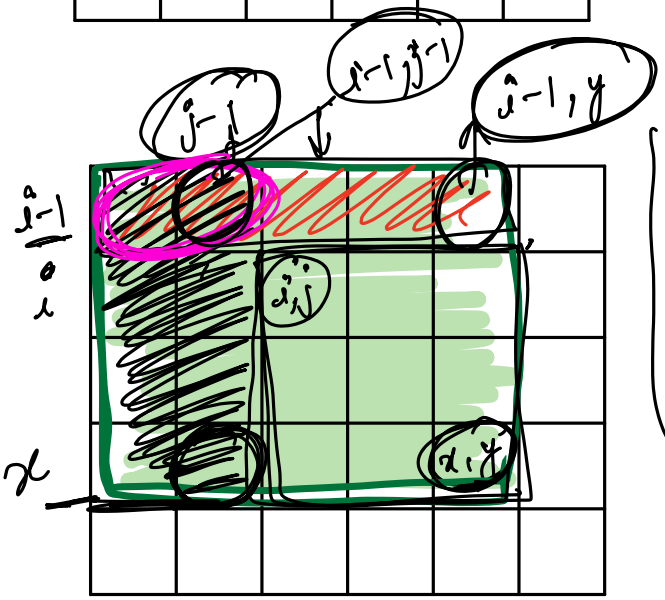
$> q*n + n^2$

$q*n*n$



$pf[i]$ = sum $0 \to i$

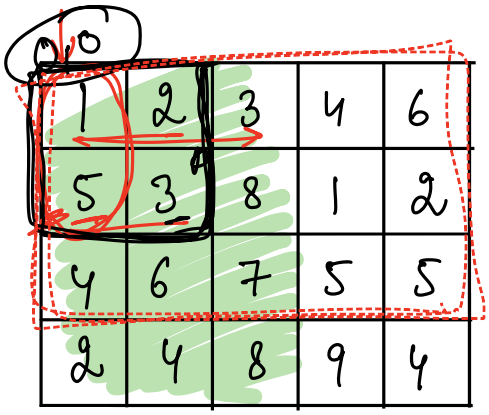
$pf[i][j]$ = sum of submatrix $(0,0) \to (i,j)$



$O(1)$

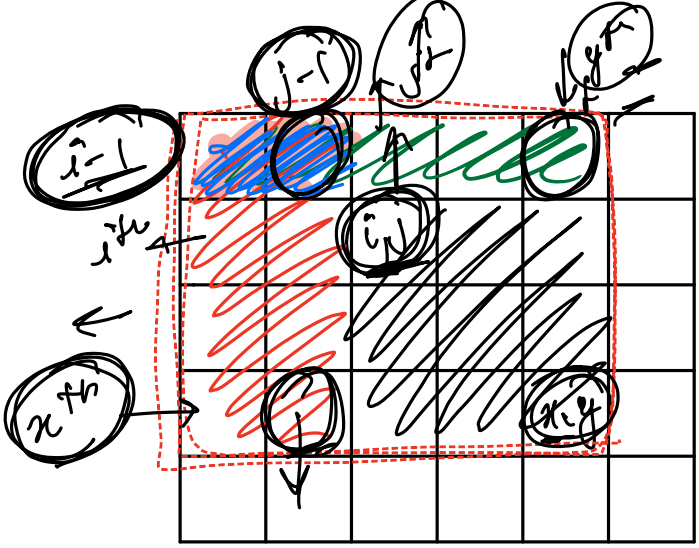
$$\boxed{pf[x][y] - pf[i-1][y] - pf[x][j-1] + pf[i-1][j-1]}$$

$+ q*1$



pf

1	3	6	10	16
5	11	22		
				0
		0		



Red - pf[x, y]

- i-1, y
- x, j-1
+ i-1, j-1

T.C = 3, 4

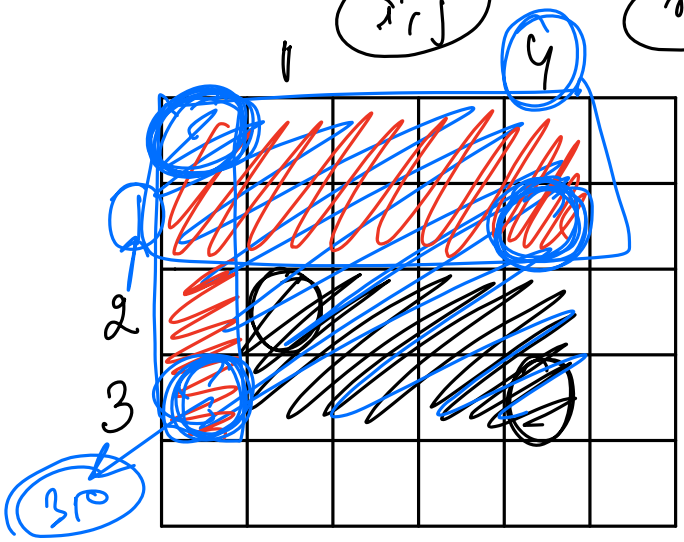
R.B 2 1

2, 1

i, j

3, 4

x, y



pf[3][4] - pf[1][4]

How to calculate prefix sum

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

Row

1	3	6	10	16
5	8	12	17	19
4	10	19	22	27
2	6	14	23	27

55

Time? → $O(n^2)$ + $q * 1$

$q + n^2$

$q n^2$

5 matrix ⇒ row wise & column wise sorted

facebook

maximum submatrix sum

-5	-4	-1
-3	-2	4
2	5	8

B.F: consider all submatrices

T.C: i, j → R.B: x, y

4 loops → $O(n^4)$

TC RB

sorted

Obs1 ⇒ $a[n][m]$ → max value

Obs2 → matrix - positive → ans? complete matrix

matrix + negative → ans? → $a[n][m]$

-5	-4	-3
~~-4~~	~~-4~~	~~-2~~
~~-3~~	~~-3~~	-1

final obs → Bottom right corner - $a[n][m]$ has to be included in your answer

T.C (i, j) R.B

for(i=0 ———)
for(j=0 ———)

TL RB
↓
prevques

0,0

-5	-4	-1
-3	-2	4
2	5	8

T.C: $O(n^2)$

TL

Prefix sum matrix

```
for( int i=0; i<n; i++)
{
    for( int j=0; j<m; j++)
    {
        TL = [i, j]
        BR = [n-1, m-1]
```

$O(1)$

}
}