Arrays - 2

1.5 hr → 2.5 - 3hr

Algo

Question: Maximum sum Subarray

A = 1 2 3 4 -10 8

1+2+3+4 = 10

(1, 2, 3, 4, 8)

A = -2 1 -3 4 -1 2 1 -5 4

Ans = 6

A = 4 3 -1 5 => $\boxed{11}$

$O(n)$

Brute Force:

Consider all subarray

subarrays : $O(n^2)$

T.C to compute sum of 1 subarray: $O(n)$

T.C: $O(n^2 \times n) = \boxed{O(n^3)}$

S.C: $O(1)$

Prefix Sum

Approach 2:

Compute Prefix Sum of array

$sum(l, r) = pref[r] - pref[l-1]$

$O(n^2)$

T.C: $n^2 \times 1$ =

S.C: O(n)

prefix sum

Every Pair (i, j)

```
for(i=0; i<n; i++) {
   for(j=i; j<n; j++) {
      sum = pre[j] - pre[i-1]
      maxm = max(sum, maxm)
   }
}
```

2+3+4-10

→ [-1] [-1,2] [-1,2,3] [-1, 2, 3, 4] [-1,2,3,4,-10]

→ [2], [2,3], [2,3,4]..

9+(-10)

Naive

Approach 3 : $O(n^2)$, O(1) space

A = -1 2 3 4 -10

max_sum = 9

curr_sum = -1

```
max_sum = INT_MIN;
for(i=0; i<n; i++) {
   curr_sum = 0;
   for(j=i; j<n; j++) {
      curr_sum += a[j];
      max_sum = max(max_sum, curr_sum);
   }
}
```

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

S.C: O(1)

Prefix

A = [1, 2, 3, 4, 5]

→ Prefixes = [1] [1,2], [1,2,3], [1,2,3,4], [1,2,3,4,5]

Subarray

Suffix: (Is not a ...

[5], [5, 4], [5, 4, 3], [5, 4, 3, 2], [5, 4, 3, 2, 1]

5
4, 5
3, 4, 5

[4], [4, -1], [4, -1, 2], [4, -1, 2, 1]

4 3 5 5

6

(4, -1, 2, 1)

Kadane's Algorithm

A =	1	-2	1	-3	4	-1	2	1	-5	4
	0	1	2	3	4	5	6	7	8	9

-> Lets assume A[i : j] is the subarray with maximum

A[i], A[i+1], A[i+2], ... A[j]

Claim : No prefix array of A[i:j] will have a negatie sum.

$\text{sum}(A[i \dots j]) = \text{sum}(A[i \dots k]) + \text{sum}(A[k+1 \dots j])$

$\text{sum}(i \dots j) = \text{sum}(i \dots k) + \text{sum}(k+1 \dots j)$

(-3)

Lets assume, we have a prefix of negative ($p>0$)

prefix $\boxed{\text{sum}(i \dots k)} = -p$

-1, -2, -3 . . .

(-1, -2, -3)

p p p

$\text{sum}(i \dots j) = -3 + \text{sum}(k+1 \dots j)$

$\boxed{\text{sum}(k+1 \dots j)} = \boxed{\text{sum}(i \dots j)} + 3$

(Max Sum)

★ Every prefix of max sum subarray has to be -ve

-1 4 2 1

Ans = (4, -1, 2, 1)
6

sum(A) = 6

A = 1 -2 1 -3 4 -1 2 1 -5 5

0 1 2 3 4 5 6 7 8 9

10

1+(-2)

max_sum = ~~-INF~~ ~~1~~ ~~4~~ ~~5~~ ~~6~~ 10
curr_sum = ~~0~~ ~~1~~ ~~0~~ ~~1~~ ~~-2~~ ~~0~~ ~~4~~ ~~3~~ ~~5~~ ~~6~~ ~~1~~ ~~5~~ 10

$\boxed{\text{ans} = 6}$

Arr = -5 -4 -2 $\boxed{-1}$

$\boxed{\text{max sum} = \cancel{-INF}\ \cancel{-5}\ \cancel{-4}\ \cancel{-2}\ -1}$

curr_sum = ~~0~~ ~~-5~~ ~~0~~ ~~-4~~ ~~0~~ ~~-2~~ ~~0~~ -1

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

Edge case

```
max_sum = INT_MIN;
curr_sum = 0;
for(i = 0; i < n; i++) {
    curr_sum += a[i];
    max_sum = max(max_sum, curr_sum);
    if(curr_sum < 0) {
        curr_sum = 0;
        start = i
    }
}
return maxSum;
```

(store start, end)
(st, end)

→ if Max Sum Subarray > 0

start = start
end = i

Question: Flip a subarray

Given a binary string with 0's and 1's

S = "1 0 0 1 0 0 1 1 1 1 0 1"

1 1 0 1 1

count_ones = 7

count_ones = 8

9

Flip all 1's → 0's

0's → 1's

ones = 10

Max no of 1's in the string

001 ⇒ 100

Brute Force:

Consider all substrings to flip

substring = $O(n^2) \times n$

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

S.C: $O(1)$

Approach 2

(1) (11) (110) (1101)

(11010).

initial_ones = 5

For i = 0 to n-1

1	1	0	1	0	1	1	0
0	1	2	3	4	5	6	7

i, j $\quad i, j$ $\quad i, j$ $\quad i, j$

ones = ~~0~~ ~~1~~ 3
zeros = ~~0~~ 1

$5-2+0=\boxed{3}$

ans = 5

$5-2+1$
$=\boxed{4}$

T.C : $O(n^2)$
S.C : $O(1)$

Efficient Approach

Observations

=) select a substring which has more 0's & and less 1's &

1 1 0 1
1 0 0 1 0

When selecting a substring:

1) 1 =) decrease my answer by 1 (-1)
2) 0 =) increase my answer by 1 (1)

Replace 1's with -1
0's with 1

S = " 1 0 0 1 0 0 1 1 1 1 0 1 "

S' = " -1 1 1 -1 1 1 -1 -1 -1 -1 1 -1 "

-) Select a substring which has max 1's

=) find subarray which has max sum =) O(n)

kadane

sum = 4 + 7

initial_one

1 1 1 0 0 0 1 1 1 1 1

0 0 1 0 0 0 1 1 1 0 0 0 0 0

initial = 3

new_ones = 8

1 2 3

1 2 3

Strength

124 = 1 + 1 2 3

Mc

Winner

1 2 3

Winner

Question:

Sum of all submatrices

$$M = \begin{bmatrix} 9 & 6 \\ 5 & 4 \end{bmatrix}_{2\times 2}$$

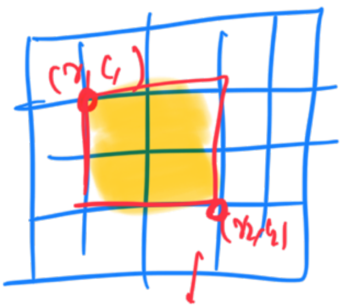
9 + 6 + 5 + 4 + 15 + 9 + 6
+ 10 + 24
= $\boxed{88}$ + 8 = $\boxed{96}$

$[9]$ 9 $[6]$ 6 $[5]$ 5 $[4]$ 4

$[9 \; 6]$ 15 $[5 \; 4]$ 9 $\begin{bmatrix} 9 \\ 5 \end{bmatrix}$ 14 $\begin{bmatrix} 6 \\ 4 \end{bmatrix}$ 10

$[9 \; 6]$

5 4] 24

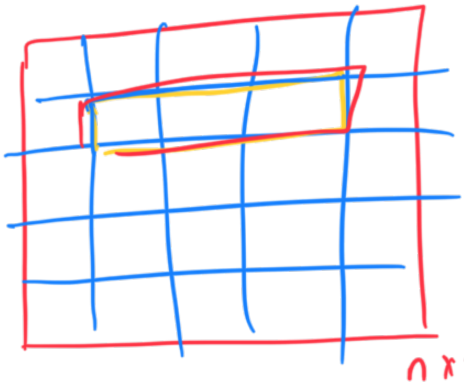


→ Top-left corner
→ Bottom-right corner

Brute Force :

Consider all submatrices

TL: (1, 1)
BR = (1, 3)



n x m

n rows
m colums

Top Left => (r_1, c_1)
Bottom Right => (r_2, c_2)

(r_1, c_2) & (r_2, c_1)

$$\boxed{\begin{matrix} r_1 \le r_2 \\ c_1 \le c_2 \end{matrix}}$$

$r_1 = 0$, $r_2 = 0 \ldots\ldots n-1$ [n]

$r_1 = 1$, $r_2 = 1 \ldots\ldots n-1$ [n-1]

$r_1 = 2$, $r_2 = 2 \ldots\ldots n-1$ [n-2]

⋮

$r_1 = n-1$, $r_2 = n-1$ [1]

No. of ways of choosing r_1 & r_2

$n + n-1 + n-2 + \ldots\ldots 1 = \frac{n(n+1)}{2}$

No of ways of choosing c_1 & c_2

$\frac{m(m+1)}{2}$

$c_1 = 0$, $c_2 = 0 \ldots\ldots m-1$

$c_1 = 1$, $c_2 = 1 \ldots\ldots n-1$

Total ways of r_1, r_2, c_1, c_2

$$\boxed{\frac{n(n+1)}{2} \times \frac{m(m+1)}{2}} = \boxed{C}$$

T.C for computing sum of 1 submatrix: $O(n \times m)$

T.C: $\frac{n \times (n+1)}{2} \times \frac{m(m+1)}{2} \times (nm)$

$: \boxed{O(n^3 m^3)}$

Better Approach:

T.C: $O(n^2 m^2 \times 1) = O(n^2 m^2)$

Efficient Approach:

Contribution of every element, Sum all of them, will I get answer

$$ar \begin{bmatrix} 9 & 6 \\ 5 & 4 \end{bmatrix}$$

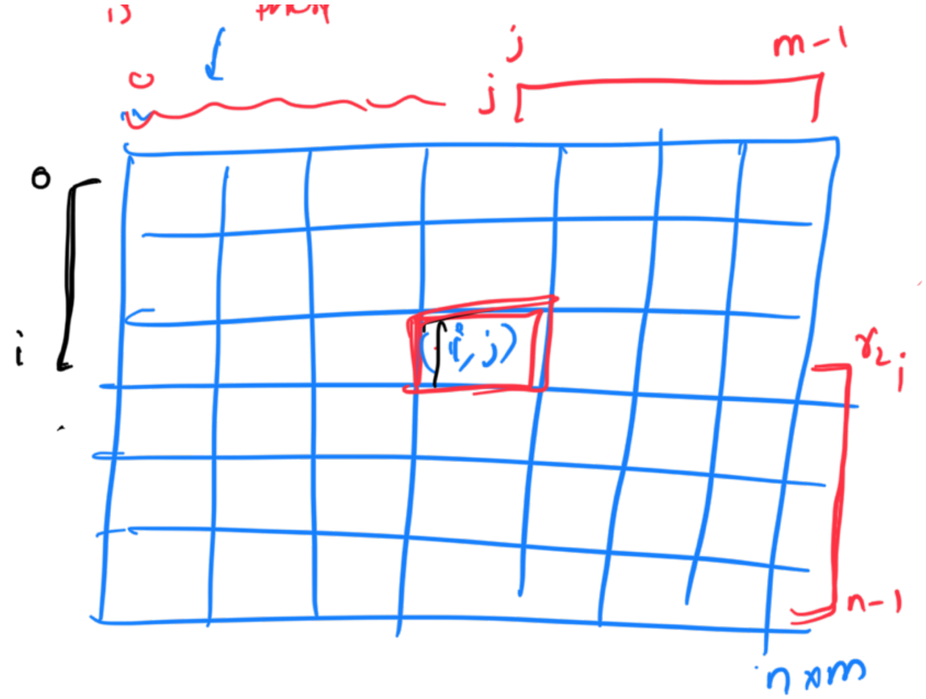
ans = # submatrices (4×9)

$\boxed{4}$

Submatrices in which 9 is there

$$9C_1 + 6C_2 + 5C_3 + 4C_4.$$

What is no. of submatrices in which a[i][j] ... there



TL, BR

(r_1, c_1)

(r_2, c_2)

[0 ... 2]

(0, 1, 2)

r_1 : [0 ... i] => i+1

r_2 : [i ... n-1] => n-i

c_1 : [0 ... j] => j+1

c_2 : [j ... m-1] = m-j

\# submatrice in which a[i][j] is a part of

(i+1) x (n-i) x (j+1) x (m-j)

```
ans = 0
for( i=0;  i< n;  i++ ) {
    for( j=0;  j<m;  j++ ) {
        ans += a[i][j] x (i+1)(n-i)(j+1) x (m-j)
    }
}
```

T.C : O(n.m)

S.C : O(1)

> 10^9

$10^9 + 7$

Queries

Module 7

1) Submatrix sum
2) Rotate a matrix by 90°

5 min

11:01

Question: Submatrix Sum Queries

Q queries

→ TL, BR
(r_1, c_1) (r_2, c_2)

A =

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

4×5

Brute Force:

(r_1, c_1) (r_2, c_2)

```
for(i=r1; i<=r2; i++) {
    for(j=c1; j<=c2; j++) {
        sum += a[i][j];
```

4
7

T.C: $O(n \times m)$

Q queries

T.C: $O(Q.N.m)$

Prefix - Sum

		l		r	
A =	3	4	1	2	-1
pref =	3	7	8	10	9

pref[i] = sum(0.. i)

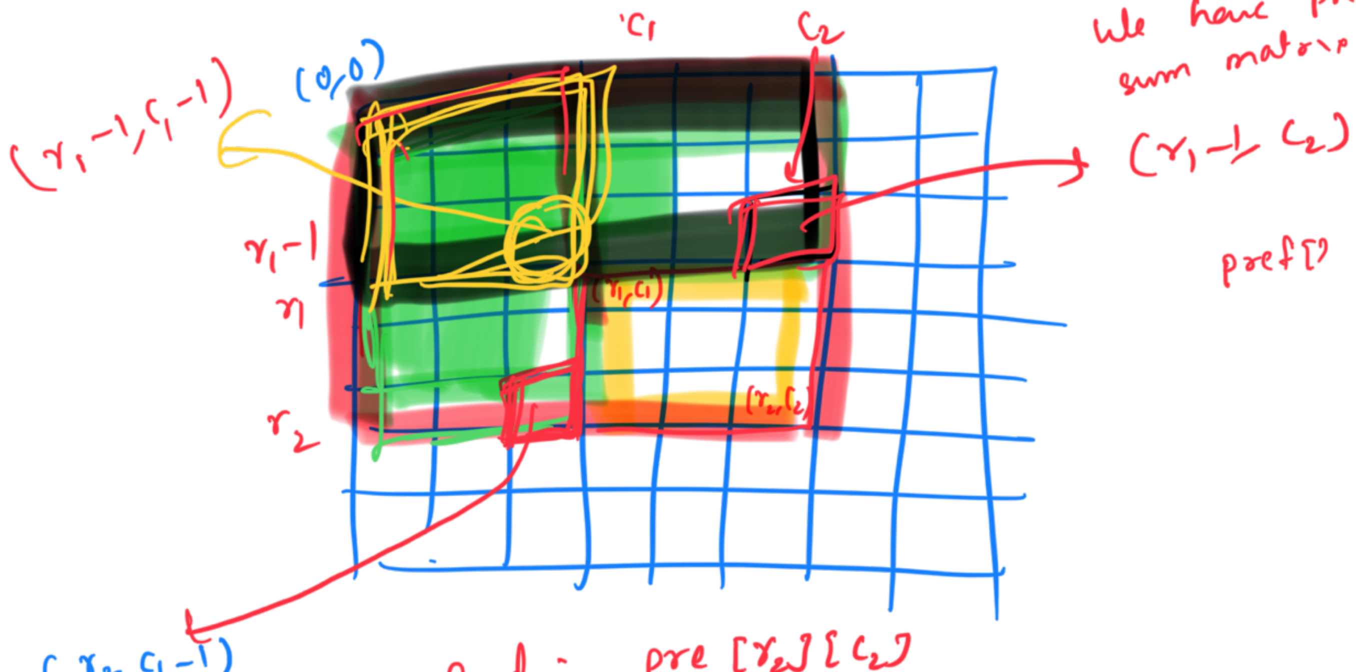
[l, r] = $\boxed{pre[r] - pre[l-1]}$

2D Prefix Sum

pre[i][j] = stores sum of submatrix from (0, 0) to (i, j)

(0, 0) → TL

(i, j) → BR



We have pref sum matrix

pref[]

Red : pre[r2][c2]

Green : pre[r2][c1-1]

Black : pre[r1-1][c2]

Yellow : pre[r1-1][c1-1]

sum = pre[r2][c2] − pre[r2][c1-1] − pre[r1-1][c2]

+ pre[r1-1][c1-1]

Query (r1, c1) (r2, c2)

For every query

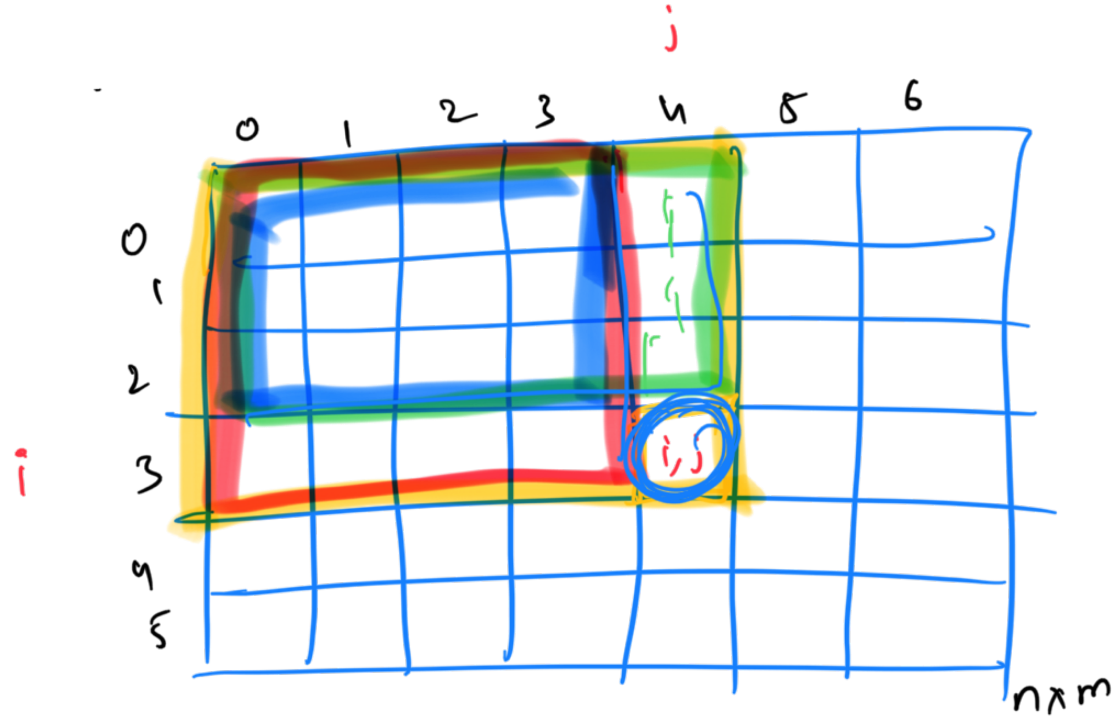
T.C : $O(1)$ for 1 query

for Q queries : $O(Q)$

prefix sum Matrix!

Build 2D Prefix Sum

$$A = \begin{bmatrix} 2 & -1 & 5 & 6 \\ -3 & 4 & 1 & 7 \\ 3 & 6 & 2 & 1 \end{bmatrix}_{3\times 4} \qquad pre = \begin{bmatrix} 2 & 1 & 6 & 12 \\ -1 & 2 & 8 & 21 \\ & & & \end{bmatrix}_{3\times 4}$$



$$pre[i][j] = pre[i][j-1] + pre[i-1][j] - pre[i-1][j-1] + a[i][j]$$

Red : $pre[i][j-1]$

Green : $pre[i-1][j]$

$pre[i-1][j-1]$

Blue : pr...

prefix sum of 1D Array

A =

3	4	1	0
2			
6			

=> Pre =

3	7	8	0
5			
11			

Generating 2D Prefix

```
for (i=1; i<n; i++) {
    for(j; j<m; j++) {
        pre[i][j] = A[i][j] + pre[i][j-1] +
                    pre[i-1][j] - pre[i-1][j-1]
    }
}
```

$\boxed{T.C : O(n \cdot m)}$

S.C : $O(n \cdot m)$

1) Generate 2D prefix

T.T.C : $\boxed{O(n \times m + Q)}$

for q queries

Question: Rotate a square matrix by 90°.

90°

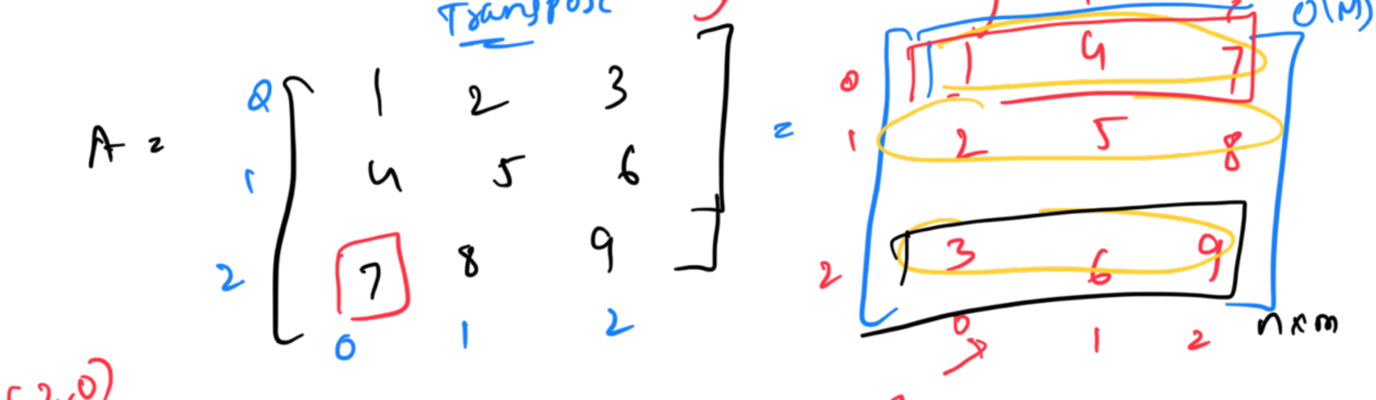
$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \Rightarrow \begin{bmatrix} 7 & 4 & 1 \\ 8 & 5 & 2 \\ 9 & 6 & 3 \end{bmatrix}$$

$O(N)$

Observations:

1) 1st row is reverse of 1st column
2) 2nd row is reverse of 2nd col

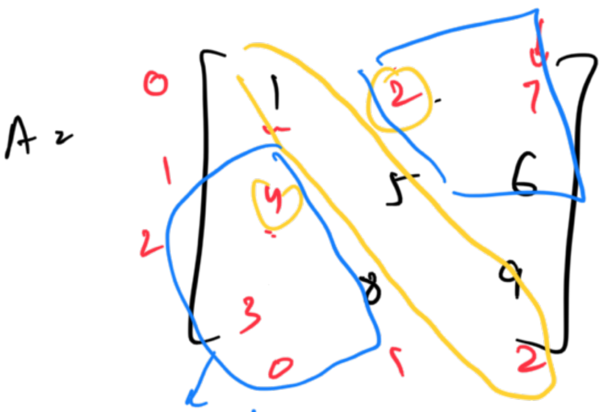
Transpose



swap(a[i][j] & a[j][i])

```
for( i=0; i<n; i++) {
    for (j=0; j<m; j++) {
        swap ( A[i][j]  A[j][i] ).
    }
}
```

upper diagonal



Lower diagonal

```
for (i=0;  i< N;  i++) {
    for (j= i+1;  j<M;  j++) {
        swap( A[i][j],  A[j][i] )
```

}

T.C: $O(m \times n)$

Algo:

1) ⟹ Find Transpose : $O(n \times m)$ ⟹ $O(1)$ space

2) Reverse all the rows → $O(1)$ space

$O(m) \times n$

$$\boxed{\begin{array}{ll} T.C: & O(n \cdot m + O \cdot m) \Rightarrow O(n \cdot m) \\ S.C: & O(1) \end{array}}$$

```
int count = 0;
for(i = N; i > 0; i = i/2) {
    for(j = 0; j < i; j++) {
        count += 1;
    }
}
```

j = 0 to j = $\frac{N}{2}$

1st iteration:
i = N

[N]

2) i = N/2

[$\frac{N}{2}$]

3) i = $\frac{N}{4}$

[$\frac{N}{4}$]

$N \to \frac{N}{2} \to \frac{N}{4} \to \frac{N}{8} \ldots 1$

1 + 0

N + $\frac{N}{2}$ + $\frac{N}{4}$ + ...

operations :

$$\Rightarrow N\left(1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \dots\right)$$

$\boxed{2}$

$\Rightarrow N \times 2$

$\Rightarrow \boxed{O(N)}$

Sum of Infinite G.P (d)

A.P = $a, \; a+d, \; a+2d, \; a+3d$

G.P = $a, \; ar, \; ar^2, \; ar^3 \dots$

$a = 1$

$r = \frac{1}{2}$

if $(r < 1)$

$$\boxed{\frac{a}{1-r}}$$

$$= \frac{1}{1-\frac{1}{2}} = \frac{1}{\left(\frac{1}{2}\right)}$$

S

$\boxed{2}$

A.P: $a, \; a+d, \; a+2d \dots$

Sum of n terms in AP : $\boxed{2a + (n-1)d}$

G.P: $a, \; ar, \; ar^2, \; ar^3 \dots$

Sum of n terms in G.P $= \boxed{\frac{a(r^n - 1)}{r-1}}$

$\boxed{{}^nC_r}$ → Combination

→ Permut

$$\boxed{\frac{n!}{r! \times (n-r)!}}$$

→ $\boxed{\text{Hashing}}$

→ Recursion

$\rightarrow \boxed{6}$

Mr

r_1, r_2, c_1, c_2

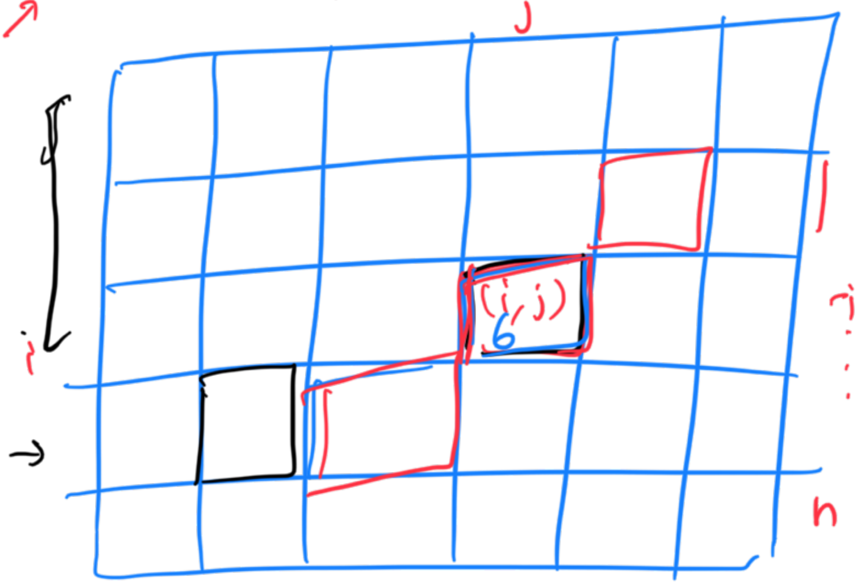
(r_1, c_1)

(r_2, c_2)

$\nearrow$ BR

TL $\nearrow$

r_1



$r_1 \le r_2$

$r_1 : [0 \dots i] \Rightarrow (i+1)$

$r_2 : [i \dots n-1] \Rightarrow (n-i)$

$c_1 :$ $(j+1)$

$c_2 :$ $(n-j)$

$\boxed{15 \text{ submatrix}}$

$\boxed{15 \times 6}$

$(a[i][j] \times \text{count})$

For (i, j)

$\text{count} = \underbrace{(i+1) \times (n-i)} \times (j+1) \times (n-j)$