Today's agenda

↳ Prefix Sum on 1D Array

↳ Prefix Sum on 2D Array

// Array recap

↳ int arr[5] : {10 20 30 40 50}

(indices 0 1 2 3 4)

print(arr[2]); → 30

//Prefix array

↳ Given arr[N], return Pf[] where

Pf(i) = Sum {arr[0], arr[1] .... arr[i]}, for all i.

Ex: arr[5]: { 5 2 7 -3 8 }

Pf[5]: { 5 7 14 11 19 }

arr[9]: { -3 6 2 4 5 2 8 -9 3 }

Pf[9]: { -3 3 5 9 14 16 24 15 18 }

//Brute force

arr[5]: { 5 2 7 -3 8 }

Pf[5]: { 5 7 14 11 }

```
int[] Prefix (int arr[N]){
    int Pf[N];
    Pf[0] = arr[0];

    for (int i=1; i<n; i++){
        int Sum=0;
        for (int j=0; j<=i; j++){
            Sum += arr[j];
        }
        Pf[i] = Sum;
    }
}
```

i		j
1	Sum=0+5 +2 =7	0 1
2	Sum=0 +5+2+7 =14	0 1 2
3	Sum=0+5 +2+7-3 =11	0 1 2 3

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

//optimal approach

$$arr[5] : \{ \overset{0}{5} \quad \overset{1}{2} \quad \overset{2}{7} \quad \overset{3}{-3} \quad \overset{4}{8} \}$$
$$pf[5] : \{ 5 \quad 7 \quad 14 \quad \}$$

↳ $pf[2] = Sum(0,2) = \underbrace{Sum(0,1)}_{pf[1]} + arr[2] = 14$

↳ $pf[3] = Sum(0,3) = \underbrace{Sum(0,2)}_{pf[2]} + arr[3] = 14-3 = 11$

$pf[i] = Sum(0, 1, \ldots i) = \underbrace{Sum(0,1, \ldots i-1)}_{pf[i-1]} + arr[i]$

⇓

$pf[i] = pf[i-1] + arr[i];$

$$arr[5] : \{ \overset{0}{5} \quad \overset{1}{2} \quad \overset{2}{7} \quad \overset{3}{-3} \quad \overset{4}{8} \}$$
$$pf[5] : \{ 5 \quad 7 \quad 14 \quad 11 \quad 19 \}$$

//PSuedo Code

```
int[] Prefix (int arr[N]){
      optimal
      int Pf[N];
      Pf[0] = arr[0];

      for(int i=1; i<n; i++){

          Pf[i] = Pf[i-1] + arr[i];

      }
      return Pf;
}
```

T.C: O(N)

Q) Given N array elements and Q queries. for each query Calculate Sum of all elements in given range.

Ex: arr[10]: {

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

}

Q = 6

L	R	ans
4	8	14
3	7	15
1	3	15
0	4	
6	9	
7	7	

//Brute force

```
void SumQuery (int arr[N], int[][]queries){
    for(int i=0; i<Q ; i++){
        int L= queries[i][0]; int R= queries[i][1]
        int Sum=0;
        for(int j=L; j<=R ; j++){
            Sum = Sum + arr[j];
        }
        Print (Sum);
    }
}
```

T.C: O(Q*N)

//optimal approach

	0	1	2	3	4	5	6	7	8	9
arr[10] : {	-2	7	3	5	6	3	9	-8	4	2 }
pf [10] : {	-2	5	8	13	19	22	31	23	27	29 }

Q = 6

L	R	ans
4	8	
3	7	
1	3	
0	4	
6	9	
7	7	

$Sum(4,8) = pf[8] - pf[3]$

$pf[8] \downarrow Sum(0,8)$ $\quad pf[3] \downarrow Sum(0,3)$

$Sum(3,7) = pf[7] - pf[2]$

$pf[7] \downarrow Sum(0,7)$ $\quad pf[2] \downarrow Sum(0,2)$

$Sum(L,R) = pf[R] - pf[L-1]$

$pf[R] \downarrow Sum(0,R)$ $\quad pf[L-1] \downarrow Sum(0,L-1)$

$Sum(\overset{L}{0},\overset{R}{4}) = pf[4] - \cancel{pf[L-1]}$

$pf[4] \downarrow Sum(0,4)$ $\quad \downarrow$ error

//Psuedo Code

```
void SumQuery (int arr[n], int[][] queries){
    int Pf[] = Pref in optimal (arr);

    for (int i=0; i<Q; i++){
        int L = queries[i][0]; int R = queries[i][1];
        if (L>0){
            int ans = Pf[R] - Pf[L-1];
            Print (ans);
        }
        else {
            int ans = Pf[R];
            Print (ans);
        }
    }
}
```

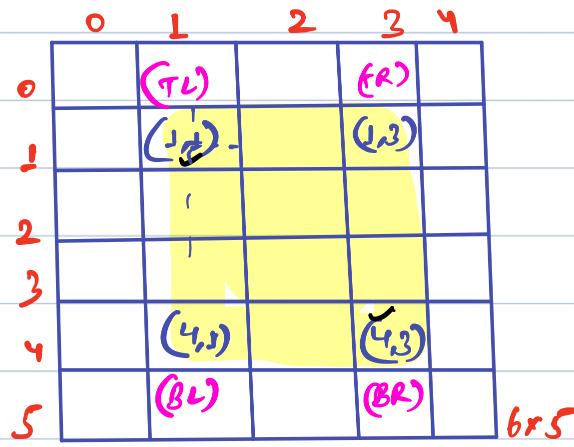
T.C: $O(N) + O(Q) = O(N+Q)$

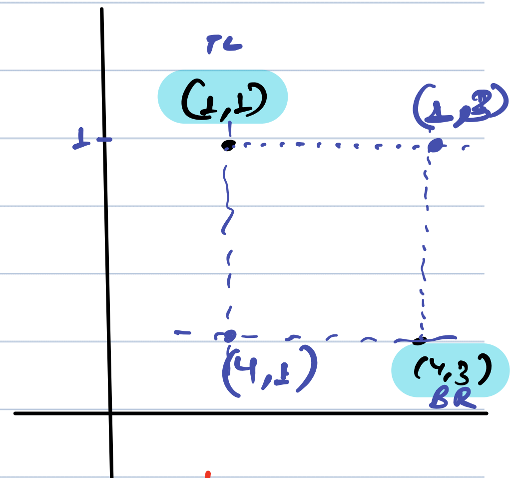
brute force → $O(Q*N)$ → $O(N^2)$

$Q \approx= N$

optimal approach → $O(Q+N)$ → $O(2N) \approx O(N)$

Representing Submatrix





↓

To represent a matrix

↓

2 corner Points

(TL & BR)

Q) Given a mat[n][m] and Q queries, for every query find submatrix sum.

	0	1	2	3	4
0	7	1	-6	3	13
1	10	5	-1	0	9
2	6	4	-3	8	11
3	13	-8	-5	12	4
4	3	2	1	9	8
5	4	3	-2	6	5

6x5
(n*m)

(Q)

TL	BR
(2,1)	(4,3): 20
(3,2)	(5,4) : 38

// Brute force
↳ for every query iterate on the submatrix and get the sum.

T.C: $O(Q * N * M)$

// Optimal approach

PSum[i] = Sum(0, i)

psum[i][j] = Sum((0,0) - (i,j))

psum[i][j] = Sum((0,0) - (i,j))

arr

	0	1	2	3	4
0	7	1	-6	3	13
1	10	5	-1	0	9
2	6	4	-3	8	11
3	13	-8	-5	12	4
4	3	2	1	9	8
5	4	3	-2	6	5

6×5 (n×m)

pf

	0	1	2	3	4
0					
1				19	
2		33			
3			23		
4					
5					

6×5 (n×m)

pf[2][1] : 33

pf[1][3] : 19

pf[3][2] : 23

// 2D Prefix Sum

arr

	0	1	2
0	a	b	c
1	d	e	f
2	g	h	i

(i) Apply row-wise Prefix Sum →

a	a+b	a+b+c
d	d+e	d+e+f
g	g+h.	g+h+i

(ii) Apply col-wise Prefix sum ↓

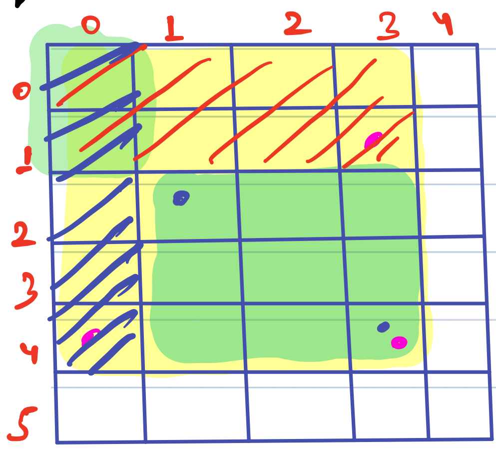
a	a+b	a+b+c
a+d	a+b+d+e	a+b+c+d+e+f
a+d+g	a+b+d+e+g+h	a+b+c+d+e+f+g+h+i

pf

	0	1	2
0	a	a+b	a+b+c
1	a+d	a+b+d+e	a+b+c+d+e+f
2	a+d+g	a+b+d+e+g+h	a+b+c+d+e+f+g+h+i

//Query solve

Pb



Q

(2,1) (TL)

(4,3) (BR)

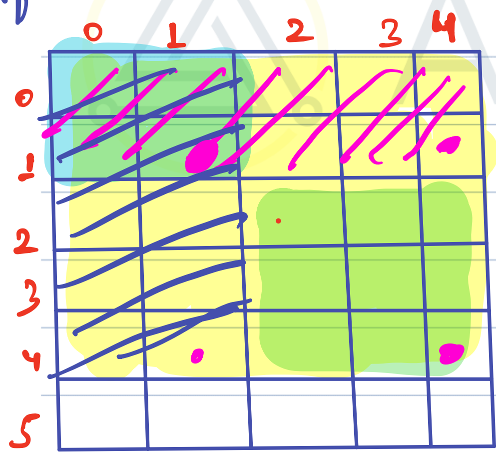
PSum[4][3] - PSum[1][3] - PSum[4][0] + PSum[1][0]

Sum[(0,0) - (4,3)]

Sum[(0,0) - (1,3)]

Sum[(0,0) - (4,0)]

Pf



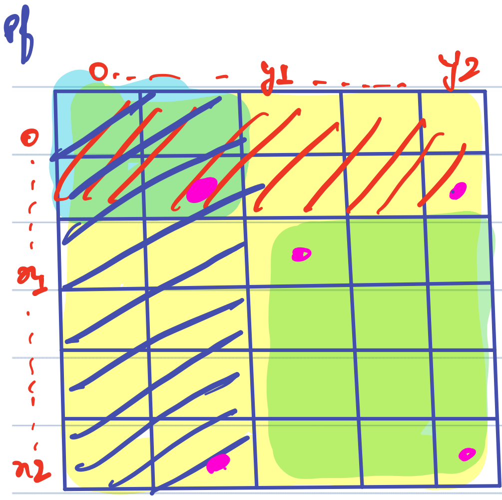
Q

(2,2) (TL)

(4,4) (BR)

Pf[4][4] - Pf[1][4] - Pf[4][1] + Pf[1][1]

Sum[(0,0) - (4,4)]



Query

$(x1, y1)$ $(x2, y2)$

$$PSum[x2][y2] - PSum[x1-1][y2] - PSum[x2][y1-1] + PSum[x1-1][y1-1]$$

//Psuedo Code

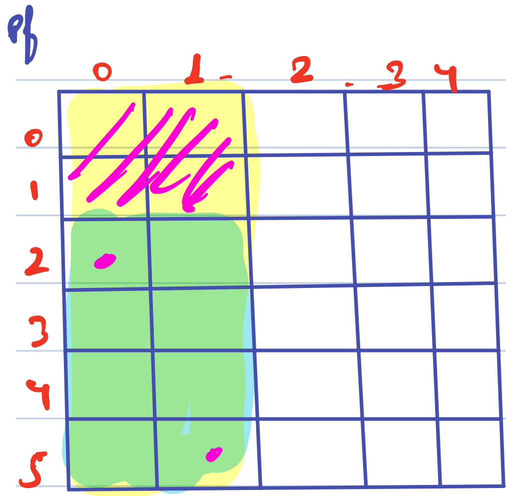
```
SubmatrixSum (int [][] mat, int [][] queries){

        int [][] PSum = func(mat);  -> HW

        for (int i=0; i< queries.length; i++){
            int x1 = queries[i][0];
            int y1 = queries[i][0];
            int x2 = queries[i][0];
            int y2 = queries[i][0];
            int Sum = 0;

            Sum += PSum[x2][y2];
            if(x1 > 0){
              Sum -= PSum[x1-1][y2];
            }
            if(y1 > 0){
              Sum -= PSum[x2][y1-1];
            }
            if(x1>0 && y1>0){
              Sum += PSum[x1-1][y1-1];
            }
        }
}
```

T.C: $O(N*M) + O(Q*1) \Rightarrow O(N*M+Q)$



Query

(2,0) (x_1, y_1) — (5,1) (x_2, y_2)

$$PSum[x_2][y_2] - PSum[x_1-1][y_2] - PSum[x_2][y_1-1] + PSum[x_1-1][y_1-1]$$

$$PSum[5][1] - PS[1][1] - PS\cancel{[5]}\cancel{[-1]} \quad PS\cancel{u[1]}\cancel{[-1]}$$