Q. Given a matrix of size N×M. Find the max submatrix sum.

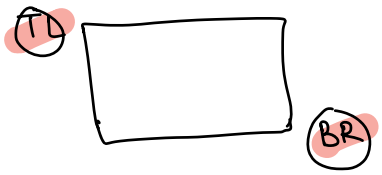
Google Quora

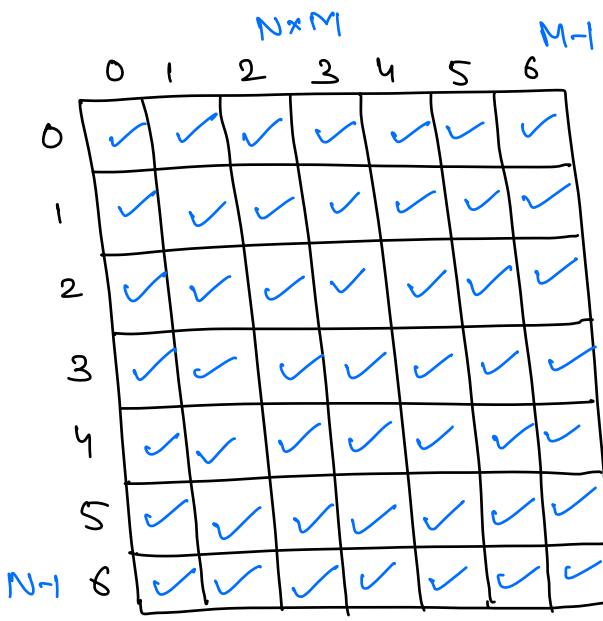
2×3

	0	1	2
0	-2	3	1
1	3	6	-3

↳ 10

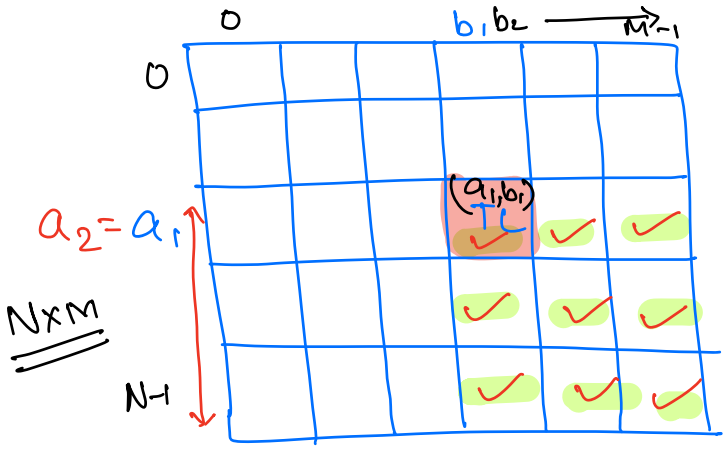
Submatrix ⇒





TL ⇒ (0,0) to (N-1, M-1)

↳ N×M



TL ⇒ (a_1, b_1)

BR ⇒ (a_2, b_2)

a_2 : a_1 to N-1

b_2 : b_1 to M-1

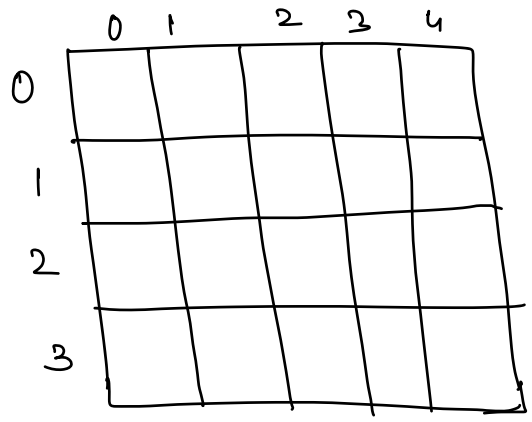
```
for(a1 = 0; a1 < N; a1++) {
    for(b1 = 0; b1 < M; b1++) {
        // TL : (a1, b1) is fixed.
        for(a2 = a1; a2 < N; a2++) {
            for(b2 = b1; b2 < M; b2++) {
                // BR : (a2, b2) is fixed
                // sum of submatrix TL, BR
                // Using PS matrix
            }
        }
    }
}
```

```
sum = PS[a2][b2] - PS[a2][b1-1]
      - PS[a1-1][b2] + PS[a1-1][b1-1]
```

TC : $O(N^2 \times M^2)$

SC : $O(N \cdot M)$ → PS matrix

} Brute force



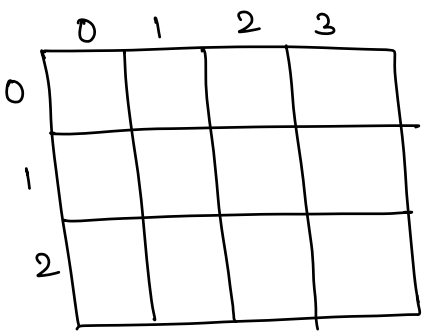
TL ⇒ (0,0)

BR ⇒ 0,0
0,1
0,2
0,3
0,4
1,0
1,1
1,2
1,3
1,4
2,0
2,1
.
.
.

Optimization :-

(I) Find the max submatrix sum where submatrix starts at row = 0 & end at row = N-1.

3×4
N×M



m ⇒ 4 + 3 + 2 + 1

$= \frac{4(4+1)}{2} \Leftarrow$

$= \frac{M(M+1)}{2}$

3×5

	0	1	2	3	4
0	-3	4	2	2	9
1	-9	-3	3	3	-3
2	-1	6	-4	4	-10

(5)

Submatrices starting at row = 0 & ending at row = N-1.

Column sums

	0	1	2	3	4
arr:	-13	7	1	9	-4

arr[i] : sum of i^{th} col

Kadane's Algo ⇒ find the max subarray sum.

(2) Find the max submatrix sum where submatrix starts at row = 0 & end at any row.

	0	1	2	3	4
0	-3	4	2	2	9
1	-9	-3	3	3	-3
2	-1	6	-4	4	-10

Start row = 0

end row = 0

-3	4	2	2	9

max subarray sum.

end row = (1)

⇒

-12	1	5	5	6

end row = 2

Ex mat[4][5]

	0	1	2	3	4
Start ⇒ 0	1	3	-10	-3	4
⇒ Start 1	(-3)	-4	2	-2	10
2	2	6	-9	6	-1
3	-4	(2)	4	-3	-8

← Submatrices starting at row = 0 and can end at any row.

mat[0][j] j → 0 to M-1

Start = 0

end = 0

arr[M]

1	3	-10	-3	4

⇐

max sum subarray.

Start = 0

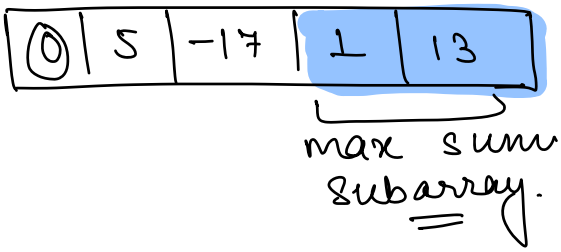
end = (1)

arr

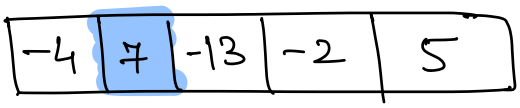
-2	-1	-8	-5	14

max sum subarray.

Start = 0 end = 2



Start = 0 end = 3



int arr[M];

ans = $-\infty$

for (start = 0; start < N; start++) { ⇒ (N)

arr[M] = {0}

for (end = start; end < N; end++) { ⇒ (N)

2M [(M) [for (j = 0; j < M; j++) {
arr[j] += mat[end][j]
}

(M) [value = maxSubArraySum(arr) → Kadane's Algo
ans = max(ans, value);

}

3×3

	0	1	2
0	1	4	9
1	-2	-8	13
2	3	6	-3

start ⇒ 0 to 2.
end ⇒ start to 2.

start = 0
end = 0

arr : | 1 | 4 | 9 |

max sum subarray = (14) = ans

end = 1

-1	-4	22

ans = 22

N * N [M + M]

⇒ TC : $O(N^2 * M)$

↳ Best Approach TC

SC : O(M)

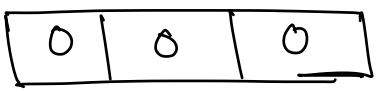
end = 2

arr ⇒ 

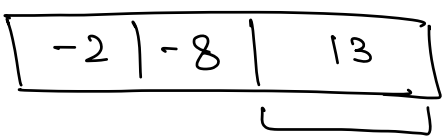
val = 23

ans = 23

Start = 1 , end ⇒ ~~0~~, 1 , 2

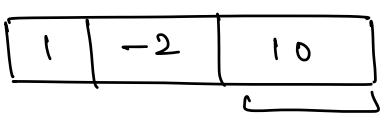


Start = 1 , end = 1



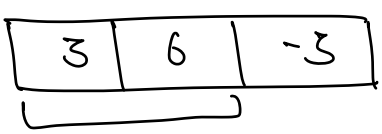
ans = 23

Start = 1 , end = 2



ans = 23

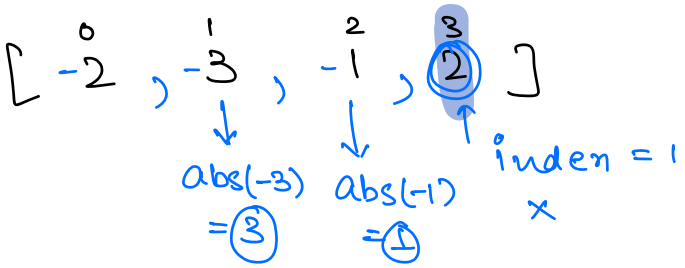
Start = 2 , end = 2



value = 9

ans = 23

Doubts



-1 * abs(-3)