{ 5 2 3 6 4 8 10 }

{ 5 7 10 _ _ _ _ }

curr = 0

for(i = 0 ; i < n; i++)

currSum += arr[i]

pref[i] = currSum

{ 5 2 3 6 4 8 10 }

0 1 2 3 4 5 6

Q = 5 { 5 7 10 16 20 28 38 }

0 1 2 3 4 5 6

2 4 -> 20 - 7 => 13

1 2

5 5

3 5

0 6

Prefix Sum arr

pre[4] - pref[3]

20 - 16 = 4

5 5

if(sp == 0) return pref[ep]

pref[ep] - pref[sp-1]

sp ep

0 0 3 5 5 2 2 3 4 1 0

0 0 3 5 5 2 2 3 4 1 0

3 2 -3 1 1 -3 -1

0 1 2 3 4 5 6 7 8 9 10

No.glass.	from	to
3	2	5
3	7	9
2	3	7
1	8	10

O(Q)

for to

pref[]

1. ∀ query

+ve -> sp

-ve -> ep

2. Prefix of the created arr

3. check if any pref[i] > cap

arr => (1 2 3 4 5 6)

0 1 2 3 4 5

Contri => [2 -1 0 2 0 -1]

[1 1 2 2 3 3]

$4 \times 2 + 5 \times 3 + 6 \times 3$

+sp

-(ep+1)

2 3

3 5

0 1

3 5

4 4

0 0

1 -1

sp ep

1 1 1 1 3 3

1 1 1 3 3 1

1 0 2 0 -2

0 1 2 3 4 5

1. Contri

2. Sort both

3. ans += Cont[i]*arr[i]

2 3 => 3 9

3 5 => 8 14

0 1 => 11 3

3 4 => 4 11

4 4 => 4 6

30 46

5 6 3 4 1 2

30142

56321

12015

41017

10305

	0	1	2	3
0	3	0	1	4
1	5	6	3	2
r_1 2	1	2	0	1
r_2 3	4	1	0	1
4	1	0	3	0

c_1 c_2

Pref sum 2D arr

	0	1	2	3
0	3	3	4	8
1	8	14	18	24
2	9	17	31	37
3	14	23	37	45
4				

A = 10 -> arr[i][j]

B = 17 -> pref[i][j-1]

C = 18 pref[i-1][j]

D = 14 pref[i-1][j-1]

1 + 31 + 24 - 19

pref[i][j] = A + B + C - D

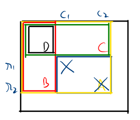
r_1 c_1 r_2 c_2

2 1 3 3 O(1)

1 0 4 1

45 - 14 - 24 + 8

15



r_1 c_1 r_2 c_2

A - B - C + D =>

(r_2,c_2) (r_2,c_1-1) (r_1-1,c_2) (r_1-1,c_1-1)