

# Fenwick Tree

arr:     0   1   2   3   4   5   6   7   8   9   10   11   12   13   14   15  
          3   2   -1   2   9   5   -8   9   3   2   1   3   -2   -3   10   7

16	1	0	0	0	0	7	16
15	1	1	1	1		10	15
14	1	1	1	0		-3	14
13	1	1	0	1		-2	13
12	1	1	0	0		3	12
11	1	0	1	1		1	11
10	1	0	1	0		2	10
9	1	0	0	1		3	9
8	1	0	0	0		4	8
7	0	1	1	1		-8	7
6	0	1	1	0		5	6
5	0	1	0	1		4	5
4	0	1	0	0		2	4
3	0	0	1	1		-1	3
2	0	0	1	0		3	2
1	0	0	0	1		2	1
0	0	0	0	0		0	0

Q. SumTill 15

Indices of

1 1111  
+  
1110

+  
1100

+  
1000

+  
0000

Q. SumTill 11

indices at

1 1011  
+  
1010  
+  
1000

+  
0000

Q. SumTill 6

0110  
+  
0100

+  
0000

```
int SumTill(int idx)
```

```
int ans = 0;
```

```
while (idx > 0)
```

```
ans += fenTree[idx];
```

```
idx -= (idx & -idx);
```

```
return ans;
```

16	1 0 0 0 0	7	16
15	1 1 1 1	10	15
14	1 1 1 0	-3	14
13	1 1 0 1	-2	13
12	1 1 0 0	3	12
11	1 0 1 1	1	11
10	1 0 1 0	2	10
9	1 0 0 1	3	9
8	1 0 0 0	4	8
7	0 1 1 1	-8	7
6	0 1 1 0	5	6
5	0 1 0 1	4	5
4	0 1 0 0	2	4
3	0 0 1 1	-1	3
2	0 0 1 0	3	2
1	0 0 0 1	2	1
0	0 0 0 0	0	0

Q. update  
Idx 6 + 5

← 1 0 0 0 0  
+5

Q. update  
Idx 3 + 2

← 1 0 0 0 0

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

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

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

← 1 0 0 0  
+5

← 0 1 0 0 0  
+2

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

← 0 1 1 0  
+5

← 0 0 1 0 0  
+2

← 0 0 0 1 1  
+2

0 0 0 1 1  
1  
0 0 1 0 0

```
void updateFrom(int idx, int val)
while (idx < n+1)
    fenTree[idx] += val;
    idx += (idx & -idx);
```

```
void buildFenTree(nums)
for (int i=0; i<n; i++)
    updateFrom(i+1, nums[i])
```

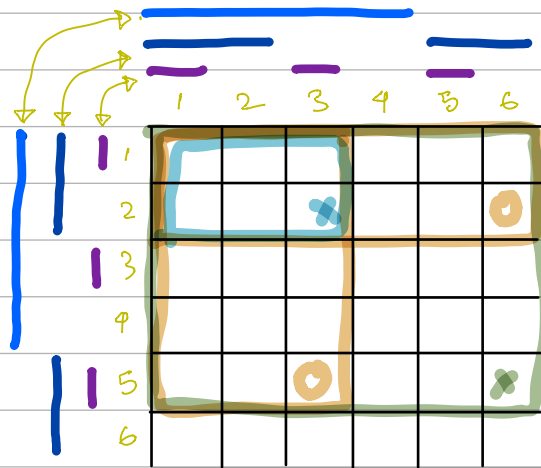
$O(n \log n)$

```
void buildFenTree(nums)
prefixSum: -
for (int i=1; i<n+1; i++)
    fenTree[i] =
    prefixSum[i] - prefixSum[i-(i&-i)]
```

$O(n)$

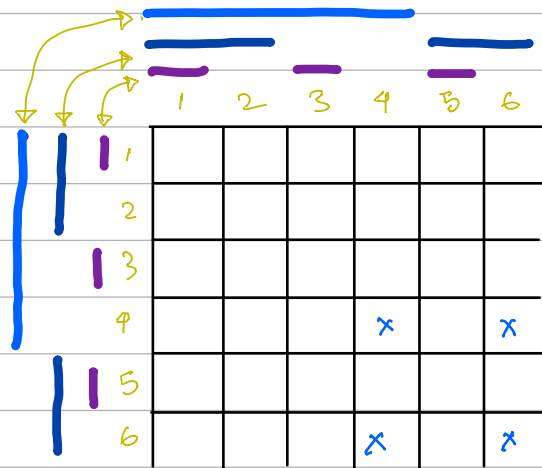
# > 2D Fenwick Tree

	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						



$$\text{RangeSum}(2,3 \text{ to } 5,6) \rightarrow \text{SumTill}(5,6) - \text{SumTill}(5,3) - \text{SumTill}(2,6) + \text{SumTill}(2,3)$$

	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						

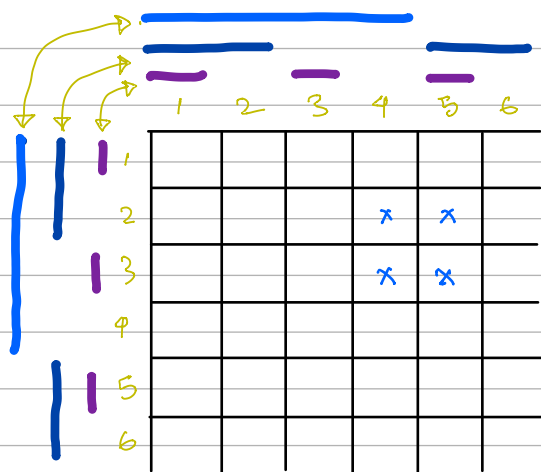


$$\text{SumTill}(6,6) \rightarrow i=110 \quad j=110 \quad i>0 \quad i-= (i \& -i) \quad j>0 \quad j-= (j \& -j)$$

$$i \rightarrow 110 \quad i \rightarrow 100 \quad i \rightarrow 0 \quad j \rightarrow 110,100,000 \quad 110,100,000 \quad 110,100,000$$

1 index arr; 0 row, col = 0.

	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						



$$\text{sumTill}(3,5) \quad 011,101 \quad 010,100 \quad 000,000$$

	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						

	1	2	3	4	5	6
1						
2	x					
3	x					
4						
5						
6						

sumTill(3, 1)

011, 001  
010, 000  
000

```

int sumTill(int x, int y)
{
    int ans = 0;
    for (; x > 0; x -= (x & -x))
        for (; y > 0; y -= (y & -y))
            ans += fenTree[x][y];
    return ans;
}

```

	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						

	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

update(1, 2)

1 0 1    1 0 2  
2 1 0    1 0 0 4  
4 1 0 0    1 0 0 0  
1 0 0 0

```

void updateFrom(int x, int y, int diff)
{
    for (; x < n+1; x += (x & -x))
        for (; y < n+1; y += (y & -y))
            fenTree[x][y] += diff;
}

```

