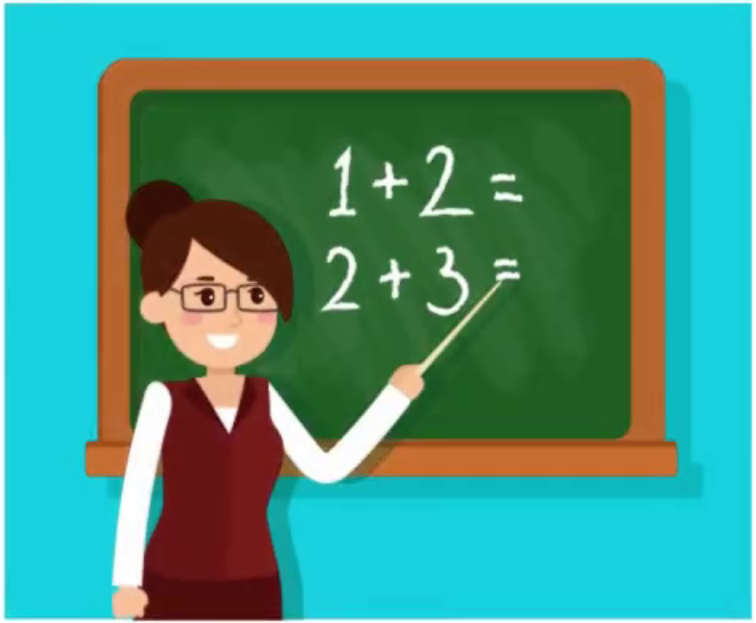
RANGED SUM

ANIMATED INTUITION

ARR[L]+=1 ARR[R]-=1

ADD FROM L TO R

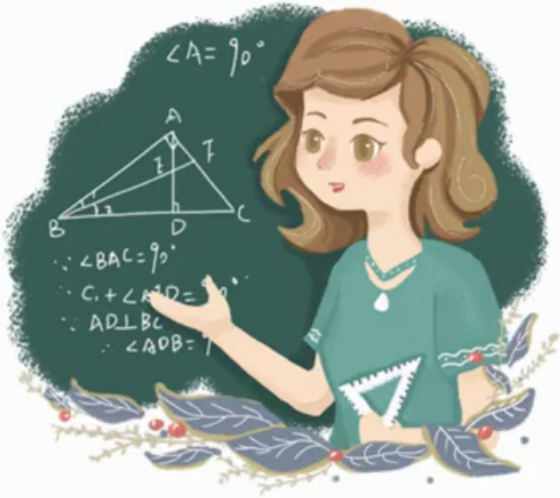


Ranged Sum

problem description :

given an array of zeros
of size n and t queries
in each query you are given
two numbers l r add one to
the numbers from l to r

print the final array



Ranged Sum

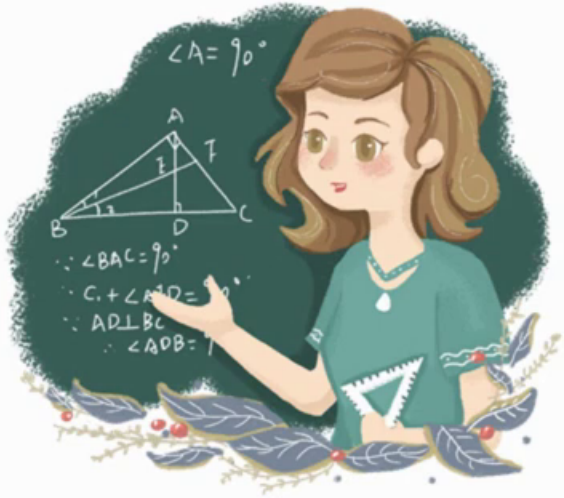
problem :

input :

```
5
2
2 4
1 4
```

output :

```
1 2 2 2 0
```



Ranged Sum

arr 0 0 0 0 0

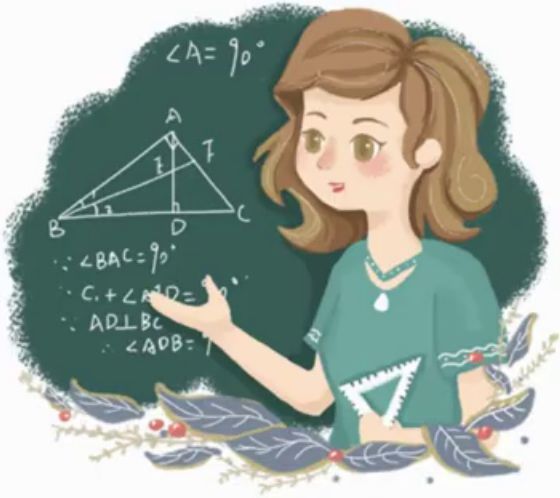
l=2 r=4

l=1 r=4

sum 1 2 2 2 0

if we used two nested for loops

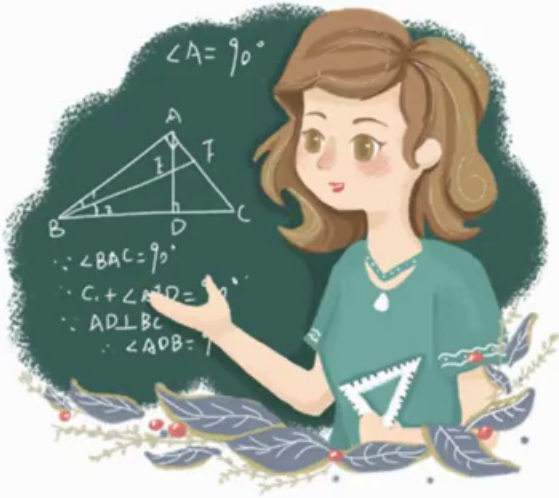
time complexity will be o(t*n)



Ranged Sum

okay the idea to solve
this problem efficiently is to

find a way to calcute the final array
by using signs you only know
l and r



Ranged Sum

we only need to know where to start and where to stop

arr 0 0 0 0 0 0

l=2 r=4

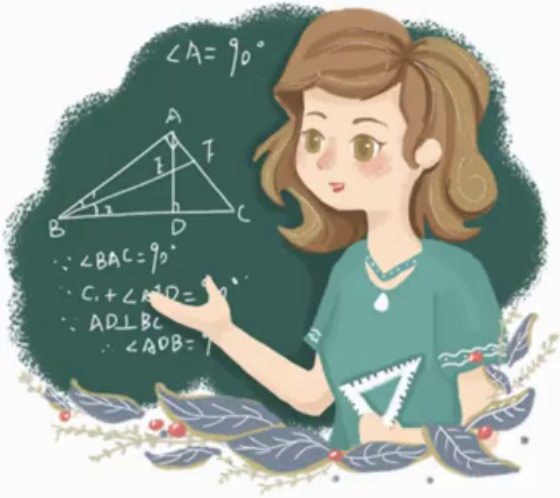
l=1 r=4

arr[l]+=1 arr[r+1]-=1

output : 1 1 0 0 0 -2

if (i){arr[i]+=arr[i-1];}

output : 1 2 2 2 2 0



Ranged Sum

n=5 but i am using 6 positions

arr 0 0 0 0 0 0

l=2 r=4

arr[l]+=1 arr[r+1]-=1

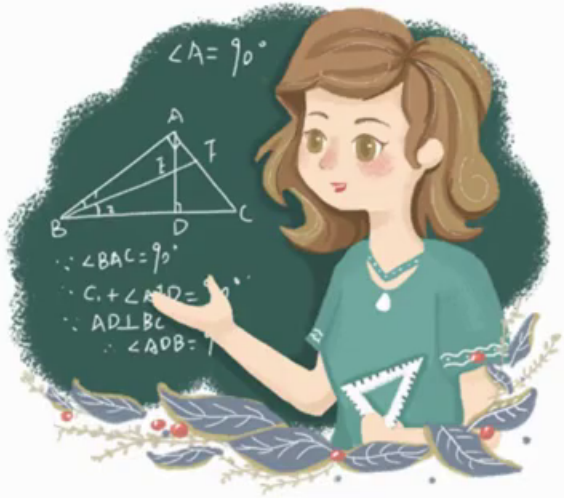
arr 0 1 0 0 -1 0

if (i){arr[i]+=arr[i-1];}

arr 0 1 1 1 0 0

1 is my starting position

and -1 is where i stop



Ranged Sum

how we improved the complexity ?

inside each query

arr[l]+=val arr[r+1]-=val o(t)

after all the queries

arr[i]+=arr[i-1] in a loop from start to end o(n)

over all take the maximum