Complexity Order

logn < sqrt(n) < n < nlogn< n*sqrt(n) < n*n < ......

For i*i<n iteration, TC => O(sqrt(n))
For i<n/2 iteration, TC => O(logn)

Range of Long-
Min: -2^63 = -9*(10^18)
Max: 2^63-1 = 9*(10^18)

Arithmetic Progression

sum = N/2*(2*a0+(n-1)*d)

Geometric Progression

sum = (a*(r^n-1))/r-1

Prefix Array
Pf[i] = pf[R]-pf[L-1] ,L>0
= pf[R] ,L=0]

Total Subarrays
n*(n+1)/2

Number of subarrays of size k
(n-k+1)

Total the sum of all subarrays
Sum += (i+1)*(n-i)*arr[i]

For Matrix[N][N], boundary in clockwise:
N-1 ➡️
N-1 ⬇️
N-1 ⬅️
N-1 ⬆️

Rotate a matrix by 90` clockwise
Transpose the matrix + reverse of each row

Sliding windows

Sum = sum -a[s-1]+a[e]

Josepher Problem
If number 2^n, then 1
Otherwise 2n+1, n=number - nearest power of 2 less than the number

Bit Manipulation

k-based to decimal/2 based number
a[n-1]*k^0+a[n-2]*k^1+a[n-3]*k^2+...+a[0]*k^(n-1)

Decimal to k-based number
```
Until n=0
{ n = n//k
  rem = n%k
  str = rem+str }
```

Properties

1> A & 1 = 0, if A is even
= 1, if A is odd

2> A & 0 = 0

3> A & A = A

4> A | 0 = A

5> A | 1 = A

6> A ^ 0 = A

7> A ^ A = 0

8> A ^ 1 = A[0->n-2 bits]+0 if odd
A[0->n-2 bits]+1 if even

9> Communicative properties

A & B = B & A

A | B = B | A

A ^ B = B ^ A

10> Associative Properties

(A & B) & C = A & (B & C)

(A | B) | C = A | (B | C)

(A ^ B) ^ C = A ^ (B ^ C)

11> 1 Left Shift => Multiplies by 2

12> 1 Right Shift => Division by 2

13> A<<n = A*(2^n) and A>>n = A/(2^n)

14> 1<<n => 2^n
15 > N | 1<<i => N ,if ith bit set
N+1<<i ,if ith bit is unset

16> N ^ 1<<i => Flips or toggle ith bit

17> N & 1<<i => 1<<i ,if ith bit is set
0 ,if ith bit is unset

Check if the ith bit is set or not

1> N & (1<<i) == 1<<i
2> N | (1<<i) == N
3> N ^ (1<<i) == N - (1<<i)
4> (N>>i)&1 == 1

Then do => N^(1<<i) OR N+(1<<i)

Negative no.s

Step1> 1's complement A = A'
Step2> 2's complement (A`+1)

MSB(Most Significant Bit) - nth bit in 8-digit binary number
MSB - 1 then -Ve
0 then +Ve

Modular Arithmatic

1> (a+b)%p = (a%p + b%p)%p
2> (a*b)%p = (a%p * b%p)%p
3> (a-b)%p = ((a%p - b%p)%p + p)%p
4> (a%p)%p = a%p
5> (a%p * b) = (a*b)%p
Fermat Theorem: a**(p-2)%p = a**(-1)%p

String

‘A’ - 65
‘Z’ - 90
‘a’ - 97
‘Z’ - 122
‘0’ - 48
‘9’ - 57

Toggling:
Caps->small: s+32
small->Caps: s-32