Theoretical Guide
Lenhadoras de Segtree

Nathália Pereira, Duda Holanda & Duda Carvalho

Contents

1 Progressions

1.1 Geometric Progression

1.1.1 General Term

$$a_1 q^{n-1}$$

1.1.2 Sum

$$\frac{a_1(q^n - 1)}{q - 1}$$

1.1.3 Infinite Sum

$$-1 < q < 1$$
$$\frac{a_1}{1 - q}$$

1.2 Arithmetic Progression

1.2.1 General Term

$$a_1 + (n-1)r$$

1.2.2 Sum

$$\frac{(a_1 + a_n)n}{2}$$

1.2.3 Sum of Second Order Arithmetic Progression

a_1 is the first element of the original progression, b_1 is the first element of the derived progression, n is the number of elements of the original progression and r is the ratio of the derived progression

$$a_1 n + \frac{(b_1 n(n-1)}{2} + \frac{rn(n-1)(n-2)}{6}$$

2 Geometry

2.1 Triangle Existence Condition and Degenerate Triangles

$$a + b \geq c$$
$$a + c \geq b$$
$$b + c \geq a$$

If it's a requirement that the triangle isn't degenerate (all of its vertices are collinear), then the existence condition is:

$$a + b > c$$
$$a + c > b$$
$$b + c > a$$

2.2 Distances

2.2.1 Euclidean

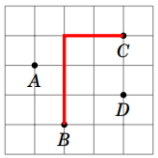
$$d(p, q) = \sqrt{(q.x - p.x)^2 + (q.y - p.y)^2}$$

2.2.2 Manhattan

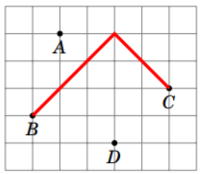
$$|p.x - q.x| + |p.y - q.y|$$

2.3 Maximum possible manhattan distance between two points given n points

Given n points, for instance:



Rotate all coordinates 45^o do that (x, y) becomes $(x + y, y - x)$, so, p becomes p' and q becomes q'.



The maximum manhattan distance is obtaining by choosing the two points that maxime:

$$max(|p'.x - q'.x|, |p'.y - q'.y|)$$

2.4 Sines Rule

$$\frac{a}{sin(\alpha)} = \frac{b}{sin(\beta)} = \frac{c}{sin(\gamma)}$$

2.5 Cossines Rule

$$a^2 = b^2 + c^2 - 2bccos(\alpha)$$

2.6 Perimeter

2.6.1 Circle

$$2\pi r$$

2.7 Areas

2.7.1 Circle

$$\pi r^2$$

2.7.2 Triangle

$$\frac{b * h}{2}$$

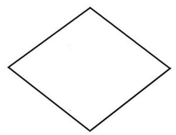
2.7.3 Square

$$l^2$$

2.7.4 Rectangle

$$hr$$

2.7.5 Rhombus



D is the biggest diagonal and d is the smallest diagonal

$$A = \frac{1}{2} * D * d$$

2.8 Volumes

2.8.1 Sphere

$$\frac{4}{3}\pi r^3$$

2.8.2 Prism

$$V = bh$$

2.8.3 Pyramid

$$\frac{bh}{3}$$

2.8.4 Cone

$$\frac{\pi r^2 h}{3}$$

2.9 Pick's Theorem

$$A = a + \frac{b}{2} - 1$$

where A is the area of the polygon, a is the number of integer points inside the polygon and b is the number of integer points in the boundary of the polygon (not counting the vertexes).

2.10 Shoelace Formula

Calculates the area of a polygon.

$$A = \frac{1}{2}|\sum_{i=1}^{n-1}(p_i \times p_{i+1}) = \frac{1}{2}|\sum_{i=1}^{n-1}(x_i y_{i+1} - x_{i+1} y_i)|$$

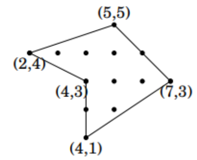
Where the points $p_1, pn, \ldots$ are in adjecent order and the first and last vertex is the same, that is, $p_1 = pn$

2.11 Boundary points

The number of integer points in the boundary of a polygon is:

$$B = v + b$$

where v is the number of vertices (integer points as well) and b is the number of integer points situated between two vertices, like in the following figure:



b can be calculated for every line connecting two points (including the line between the last and the first point) as follows:

$$boundary\_points(p, q) = \begin{cases} |p.y - q.y| - 1 & \text{p.x = q.x} \\ |p.x - q.x| - 1 & \text{p.y = q.y} \\ gcd(|p.x - q.x|, |p.y - q.y|) - 1 & \end{cases}$$

3 Misc

3.1 Check for overflow

Returns false if there is no overflow and true if there is overflow. The variable v stores the result of the operation.

```
long long v;
__builtin_add_overflow(a, b, v);
cout << v;

__builtin_sub_overflow(a, b, v);
__builtin_mul_overflow(a, b, v);
```

3.2 Input by file

freopen("input.txt","r",stdin);
freopen("output.txt","w",stdout);

4 Number Theory

$$(a + b) \bmod m = (a \bmod m + b \bmod m) \bmod m$$

$$(a - b) \bmod m = (a \bmod m - b \bmod m) \bmod m$$

$$(a \times b) \bmod m = ((a \bmod m) \times (b \bmod m)) \bmod m$$

$$a^b \bmod m = (a \bmod m)^b \bmod m$$

$$a \equiv b \pmod{m} \iff (b-a)|m$$

$$\gcd(a_1, a_2, a_3, a_4) = \gcd(a_1, \gcd(a_2, \gcd(a_3, a_4)))$$
$$\text{lcm}(a,b) \times \gcd(a,b) = a \times b$$
$$\text{lcm}(a,b) = \frac{a \times b}{\gcd(a,b)} = \frac{a}{\gcd(a,b)} \times b$$
$$gcd(a,b) = b?gcd(b, a\%b) : a$$

4.1 Fermat's Theorems

Let P be a prime number and a an integer, then:

$$a^p \equiv a \pmod{p}$$

$$a^{p-1} \equiv 1 \pmod{p}$$

Lemma: Let p be a prime number and a and b integers, then:

$$(a+b)^p \equiv a^p + b^p \pmod{p}$$

Lemma: Let p be a prime number and a an integer. The inverse of a modulo p is a^{p-2}:

$$a^{-1} \equiv a^{p-2} \pmod{p}$$

4.2 Product of divisors

Given the prime factorization

$$n = p_1^{e_1}.p_2^{e_2}.p_3^{e_3}$$

The product of divisors of n is

$$p(n) = n^{d(n)/2}$$

$$p(n) = (p_1^{e_1}.p_2^{e_2}.p_3^{e_3})^{(e1+1).(e2+1).(e3+1)/2}$$

$$p(n) = p_1^{e_1.(e_1+1)(e2+1).(e3+1)/2}.p_2^{e_2.(e_1+1)(e2+1).(e3+1)/2}.p_3^{e_3.(e_1+1)(e2+1).(e3+1)/2}$$

For any e_i, it is guaranteed that either e_i or $e_i + 1$ will be divisible by 2.

When calculating the exponent $e_1.(e_1+1)(e2+1).(e3+1)/2$, get it % MOD - 1, from Fermat's Theorem.

4.3 Sum of divisors

Given the prime factorization

$$n = p_1^{e_1}.p_2^{e_2}.p_3^{e_3}$$

The sum of divisors of n is

$$\phi(n) = \frac{p_1^{e_1+1}-1}{p_1-1}.\frac{p_2^{e_2+1}-1}{p_2-1}.\frac{p_3^{e_3+1}-1}{p_3-1}$$

4.4 Sum of digits of n in base b

$$f(n,b) = \begin{cases} n & n < b \\ f\left(n, \left\lfloor \frac{n}{b} \right\rfloor + (n \mod b)\right) & n \geq b \end{cases}$$

4.5 Number of digits of n in base b

If

$$\sqrt[k]{n} < b$$

then n has k or less digits when written in base b.

4.6 Number of divisors

Given the prime factorization

$$n = p_1^{e_1}.p_2^{e_2}.p_3^{e_3}$$

The number of divisors of n is

$$d(n) = (e1+1) * (e2+1) * (e3+1)$$

4.7 Approximation of Number of Divisors

The number of divisors of n is about $\sqrt[3]{n}$.

n	6	60	360	5040	55440	720720	4324320	21621600
$d(n)$	4	12	24	60	120	240	384	576

4.8 Prime counting function - $\pi(x)$

Expected to have $\frac{x}{\log x}$ primes within $[1, x]$. The prime counting function is asymptotic to $\frac{x}{\log x}$, by the prime number theorem.

x	10	10^2	10^3	10^4	10^5	10^6	10^7	10^8
$\pi(x)$	4	25	168	1 229	9 592	78 498	664 579	5 761 455

4.9 K leading digits of n!

A similar idea can be used to calculate the first digits of exponentiation.

$$\log_{10} n! = \log_{10}(1 \times 2 \times 3 \times ... \times n) = \log_b 1 + \log_{10} 2 + \log_{10} 3 + ... + \log_{10} n$$

Decimal part:

$$q = \log_{10} n! - (int) \log_{10} n!$$

Leading digits:

$$b = pow(10, q)$$

```
// Shift decimal point k-1 times
for ( int i = 0; i < k - 1; i++ ) {
    b *= 10;
}
```

$$leadingdigits = \lfloor b \rfloor$$

4.10 Number of digits of n! in base b

$$\lfloor \log_b n! \rfloor + 1 = \lfloor \log_b (1 \times 2 \times 3 \times ... \times n) \rfloor + 1 = \lfloor \log_b 1 + \log_b 2 + \log_b 3 + ... + \log_b n \rfloor + 1$$

5 Bitwise

5.1 Number of bits on

```
__builtin_popcount(x)
```

5.2 Count leading zeros

```
__builtin_clz(z)
__builtin_clzll(z)
```

5.3 MSB

```
32 - _builtin_clz(x)
64 - __bultin_clzll(x)
```

5.4 Count trailing zeros

```
__builtin_ctz(x)
__builtin_ctzll(x)
```

5.5 LSB

```
__builtin_ffs(X)
```

5.6 NOT

A	X
0	1
1	0

5.7 AND

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

5.8 OR

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

5.9 NAND

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

5.10 NOR

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

5.11 XOR

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

5.12 XNOR

A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

5.13 Turn bit on or off

Turn on bit i x |= (1 << i)
Turn off bit i x &= ~(1 << i)

5.14 Check if bit is on or off

Check if bit is on x & (1 << i)
Check if bit is off !(x & (1 << i))

5.15 XOR from 1 to n

$$f(n) = \begin{cases} n & n \equiv 0 \pmod 4 \\ 1 & n \equiv 1 \pmod 4 \\ n+1 & n \equiv 2 \pmod 4 \\ 0 & n \equiv 3 \pmod 4 \end{cases}$$

6 Constants

```
LLINF = 0x3f3f3f3f3f3f3f3fLL
```

```
PI = acos(-1)
```

7 Identities

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2} \qquad \sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} \qquad \sum_{i=1}^{n} i^3 = \left(\frac{n(n+1)}{2}\right)^2$$

$$\sum_{i=1}^{n} \frac{1}{i} \approx \log n \qquad \sum_{i=0}^{\infty} \frac{1}{2^i} = 2$$

8 C++

8.1 Ordered set and multiset

```
typedef tree<pair<ll , ll >, null_type , less<pair<ll , ll >>,
rb_tree_tag , tree_order_statistics_node_update> ordered_set ;
```

To change to multiset switch equal to less_equal.

8.2 Priority Queue

```
template<class T> using min_priority_queue =
priority_queue<T, vector<T>, greater<T>>;
```

9 Basic Math

9.1 Recurring Decimal

To find whether a fraction in its most simple form is a recurring decimal, find the prime factors of the denominator. If there are any prime factors other than 2 and 5 then the fraction is a recurring decimal.

9.2 Logarithm

$$\log_b mn = \log_b m + \log_b n \qquad log_b \frac{m}{n} = \log_b m - \log_n n \qquad \log_b n^p = p \log_b n$$

$$\log_b \sqrt[q]{n} = \frac{1}{q} \log_b n \qquad \log_b n = \log_a n \log_b a \qquad b^{\log_b k} = k$$

$$\log_b a = \frac{\log_c a}{\log_c b} \qquad \log_b a = \frac{1}{\log_a b} \qquad \log_b a \ \log_a c = \log_b c$$

$$\log_b 1 = 0 \qquad \log_b b = 1$$

9.3 Divisibility Criteria

9.3.1 2

The last digit is either 0, 2, 4, 6 or 8

9.3.2 3

The sum of the digits is also divisible by 3

9.3.3 4

The last two digits form a number that is divisble by 4

9.3.4 5

The last digit is either 0 or 5

9.3.5 6

It has to be divisible by both 2 and 3

9.3.6 7

The subtraction of the number formed without the last digit and the last digit times 2 is also divisible by 7

9.3.7 8

The last three digits form a number that is divisble by 8

9.3.8 9

The sum of the digits is also divisible by 9

10 Combinatorics

10.1 Catalan Numbers

The Catalan number C_n equals the number of valid parenthesis expressions that consist of n left parentheses and n right parentheses.

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$

Catalan numbers are also related to trees:

- there are C_n binary trees of n nodes
- there are C_{n-1} rooted trees of n nodes.

10.1.1 K-th convolution of Catalan

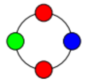
Finds the count of balanced parentheses sequences consisting of n+k pairs of parentheses where the first k symbols are open brackets.

$$C_k = \frac{k+1}{n+k+1}\binom{2n+k}{n}$$

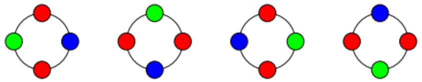
10.2 Burnside's Lemma

Burnside's lemma can be used to count the number of combinations so that only one representative is counted for each group of symmetric combinations. For example, if we have a necklace with different colored pearls and we want to know how many different combinations we can make.

For example, if we have a necklace colored like this:



This variations are the same if we consider that we can rotate the necklace:



When the number of steps is k, the number of necklaces that remain the same are:

$$m^{gcd(k,n)}$$

The number of diferent combinations for m colors and a necklace of size n is

$$\sum_{i=0}^{n-1} \frac{m^{gcd(i,n)}}{n}$$

So a necklace of length 4 with 3 colors has

$$\frac{3^4 + 3 + 3^2 + 3}{4} = 24$$