Theoretical Guide
Lenhadoras de Segtree

Nathália Pereira, Duda Holanda & Duda Carvalho

Contents

1 Number Theory

$$(a+b) \bmod m = (a \bmod m + b \bmod m) \bmod m$$
$$(a-b) \bmod m = (a \bmod m - b \bmod m) \bmod m$$
$$(a \times b) \bmod m = ((a \bmod m) \times (b \bmod m)) \bmod m$$
$$a^b \bmod m = (a \bmod m)^b \bmod m$$
$$a \equiv b \pmod{m} \iff (b-a)|m$$

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

1.1 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}:

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

1.2 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

1.3 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.

1.4 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

1.5 Sum of digits of n in base b

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

1.6 Number of digits of n! in base b

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

2 C++

2.1 Priority Queue

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

2.2 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.

3 Bitwise

3.1 NOT

A	X
0	1
1	0

3.2 AND

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

3.3 OR

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

3.4 NAND

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

3.5 NOR

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

3.6 XOR

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

3.7 XNOR

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

3.8 XOR from 1 to n

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

3.9 Number of bits on

```
__builtin_popcount(x)
```

3.10 Count leading zeros

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

3.11 MSB

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

3.12 Count trailing zeros

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

3.13 LSB

```
__builtin_ffs(X)
```

3.14 Turn bit on or off

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

3.15 Check if bit is on or off

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

4 Constants

```
LLINF = 0x3f3f3f3f3f3f3f3fLL
```

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

5 Progressions

5.1 Geometric Progression

5.1.1 General Term

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

5.1.2 Sum

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

5.1.3 Infinite Sum

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

5.2 Arithmetic Progression

5.2.1 General Term

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

5.2.2 Sum

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

5.2.3 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

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

6 Basic Math

6.1 Logarithm

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

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

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

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

6.2 Divisibility Criteria

6.2.1 2

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

6.2.2 3

The sum of the digits is also divisible by 3

6.2.3 4

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

6.2.4 5

The last digit is either 0 or 5

6.2.5 6

It has to be divisible by both 2 and 3

6.2.6 7

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

6.2.7 8

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

6.2.8 9

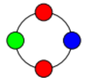
The sum of the digits is also divisible by 9

7 Combinatorics

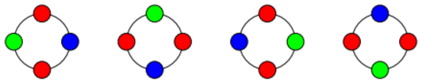
7.1 Burnside's Lemma

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

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

8 Misc

8.1 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);
```

8.2 Input by file

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

9 Geometry

9.1 Triangle Existence Condition

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

9.2 Distances

9.2.1 Euclidean

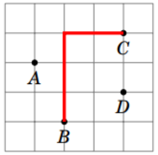
$$d(p,q) = \sqrt{(q.x - p.x)^2 + (q.y - p.y)^2}$$

9.2.2 Manhattan

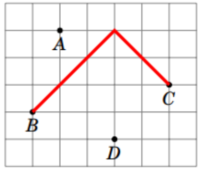
$$|p.x - q.x| + |p.y - q.y|$$

9.3 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:

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

9.4 Sines Rule

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

9.5 Cossines Rule

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

9.6 Pick's Theorem

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

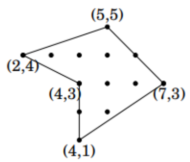
where A is the area of the polygon, a is the number of integer points inside the polygon and b is the number of integer points in the boundary of the polygon

9.7 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:

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

9.8 Perimeter

9.8.1 Circle

$$2\pi r$$

9.9 Areas

9.9.1 Circle

$$\pi r^2$$

9.9.2 Triangle

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

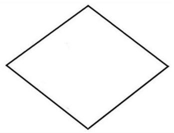
9.9.3 Square

$$l^2$$

9.9.4 Rectangle

$$hr$$

9.9.5 Rhombus



D is the biggest diagonal and d is the smallest diagonal

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

9.10 Volumes

9.10.1 Sphere

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

9.10.2 Prism

$$V = bh$$

9.10.3 Pyramid

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

9.10.4 Cone

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

9.11 Shoelace Formula

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

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

10 Identities

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

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