# Algorithm Library

magic::team.getname()

South China Normal University

August 29, 2024

# Contents

# 头文件

## DEBUG 头

```cpp
#include <bits/stdc++.h>
using namespace std;
using i64=long long;
using i128=__int128;

namespace DBG
{
    template <class T>
    void _dbg(const char *f,T t) { cerr<<f<<'='<<t<<'\n'; }

    template <class A,class... B>
    void _dbg(const char *f,A a,B... b)
    {
        while (*f!=',') cerr<<*f++;
        cerr<<'='<<a<<",";
        _dbg(f+1,b...);
    }

    template <class T>
    ostream& operator << (ostream& os,const vector<T> &v)
    {
        os<<"[ ";
        for (const auto &x:v) os<<x<<", ";
        os<<"]";
        return os;
    }

    #define dbg(...) _dbg(#__VA_ARGS__, __VA_ARGS__)
}

using namespace DBG;
```

## __int128 输出流

```cpp
ostream &operator << (ostream &os,i128 n)
{
    string s;
    bool neg=n<0;
    if (neg) n=-n;
    while (n)
    {
        s+='0'+n%10;
        n/=10;
    }
    if (neg) s+='-';
    reverse(s.begin(),s.end());
    if (s.empty()) s+='0';
    return os<<s;
}
```

## 常用数学函数

```cpp
i64 ceilDiv(i64 n,i64 m)
{
    if (n>=0) return (n+m-1)/m;
    else return n/m;
}

i64 floorDiv(i64 n,i64 m)
{
    if (n>=0) return n/m;
    else return (n-m+1)/m;
}

i128 gcd(i128 a,i128 b)
{
    return b?gcd(b,a%b):a;
}
```

## 数学

### 欧拉筛

```cpp
vector<int> minp,primes;

void sieve(int n)
{
    minp.assign(n+1,0);
    primes.clear();
    for (int i=2;i<=n;i++)
    {
        if (!minp[i])
        {
            minp[i]=i;
            primes.push_back(i);
        }
        for (auto p:primes)
        {
            if (i*p>n) break;
            minp[i*p]=p;
            if (p==minp[i]) break;
        }
    }
}
```

### 取模类（MInt）

```cpp
template <class T>
constexpr T power(T a,i64 b)
{
    T res=1;
    for (;b;b>>=1,a*=a)
        if (b&1) res*=a;
    return res;
}

template <int P>
struct MInt
{
    int x;
    constexpr MInt():x{} {}
    constexpr MInt(i64 x):x{norm(x%getMod())} {}

    static int Mod;
    constexpr static int getMod()
    {
        if (P>0) return P;
        else return Mod;
    }

    constexpr static void setMod(int Mod_) { Mod=Mod_; }

    constexpr int norm(int x) const
    {
        if (x<0) x+=getMod();
        if (x>=getMod()) x-=getMod();
        return x;
    }

    constexpr int val() const { return x; }

    explicit constexpr operator int () const { return x; }

    constexpr MInt operator - () const
    {
        MInt res;
        res.x=norm(getMod()-x);
        return res;
    }

    constexpr MInt inv() const
    {
        assert(x!=0);
        return power(*this,getMod()-2);
    }

    constexpr MInt &operator *= (MInt rhs) &
    {
        x=1ll*x*rhs.x%getMod();
        return *this;
    }

    constexpr MInt &operator += (MInt rhs) &
    {
        x=norm(x+rhs.x);
        return *this;
    }

    constexpr MInt &operator -= (MInt rhs) &
    {
        x=norm(x-rhs.x);
        return *this;
    }

    constexpr MInt &operator /= (MInt rhs) &
    {
        return *this*=rhs.inv();
    }

    friend constexpr MInt operator * (MInt lhs,MInt rhs)
    {
        MInt res=lhs;
        res*=rhs;
        return res;
    }

    friend constexpr MInt operator + (MInt lhs,MInt rhs)
    {
        MInt res=lhs;
        res+=rhs;
        return res;
    }

    friend constexpr MInt operator - (MInt lhs,MInt rhs)
    {
        MInt res=lhs;
        res-=rhs;
        return res;
    }

    friend constexpr MInt operator / (MInt lhs,MInt rhs)
    {
        MInt res=lhs;
        res/=rhs;
        return res;
    }

    friend constexpr istream &operator >> (istream &is,MInt &a)
    {
        i64 v;
        is>>v;
        a=MInt(v);
        return is;
    }

    friend constexpr ostream &operator << (ostream &os,const MInt &a) { return os<<a.val(); }
```

```cpp

    friend constexpr bool operator == (MInt lhs,MInt rhs) { return lhs.val()==rhs.val(); }

    friend constexpr bool operator != (MInt lhs,MInt rhs) { return lhs.val()!=rhs.val(); }
};

template<>
int MInt<0>::Mod=1;

template<int V,int P>
constexpr MInt<P> CInv=MInt<P>(V).inv();
```