泡泡猿 ACM 模板

Rand0w & REXWIND & Dallby

2021 年 9 月 29 日

目录

1 头文件

1.1 头文件 (Rand0w)

```
1  #include <bits/stdc++.h>
2  //#include <bits/extc++.h>
3  //using namespace __gnu_pbds;
4  //using namespace __gnu_cxx;
5  using namespace std;
6  #pragma optimize(2)
7  //#pragma GCC optimize("Ofast,no-stack-protector")
8  //#pragma GCC target("sse,sse2,sse3,ssse3,sse4,popcnt
       ,abm,mmx,avx,avx2,tune=native")
9  #define rbset(T) tree<T,null_type,less<T>,rb_tree_tag
       ,tree_order_statistics_node_update>
10 const int inf = 0x7FFFFFFF;
11 typedef long long ll;
12 typedef double db;
13 typedef long double ld;
14 template<class T>inline void MAX(T &x,T y){if(y>x)x=y
       ;}
15 template<class T>inline void MIN(T &x,T y){if(y<x)x=y
       ;}
16 namespace FastIO
17 {
18 char buf[1 << 21], buf2[1 << 21], a[20], *p1 = buf, *
       p2 = buf, hh = '\n';
19 int p, p3 = -1;
20 void read() {}
21 void print() {}
22 inline int getc()
23 {
24 return p1 == p2 && (p2 = (p1 = buf) + fread(buf, 1, 1
       << 21, stdin), p1 == p2) ? EOF : *p1++;
25 }
26 inline void flush()
27 {
28 fwrite(buf2, 1, p3 + 1, stdout), p3 = -1;
29 }
30 template <typename T, typename... T2>
31 inline void read(T &x, T2 &... oth)
32 {
33 int f = 0;x = 0;char ch = getc();
34 while (!isdigit(ch)){if (ch == '-')f = 1;ch = getc()
       ;}
35 while (isdigit(ch)){x = x * 10 + ch - 48;ch = getc()
       ;}
36 x = f ? -x : x;read(oth...);
37 }
38 template <typename T, typename... T2>
39 inline void print(T x, T2... oth)
40 {
41 if (p3 > 1 << 20)flush();
42 if (x < 0)buf2[++p3] = 45, x = -x;
43 do{a[++p] = x % 10 + 48;}while (x /= 10);
44 do{buf2[++p3] = a[p];}while (--p);
45 buf2[++p3] = hh;
46 print(oth...);
47 }
48 } // namespace FastIO
49 #define read FastIO::read
50 #define print FastIO::print
51 #define flush FastIO::flush
52 #define spt fixed<<setprecision
53 #define endll '\n'
54 #define mul(a,b,mod) (__int128)(a)*(b)%(mod)
55 #define pii(a,b) pair<a,b>
56 #define pow powmod
57 #define X first
58 #define Y second
59 #define lowbit(x) (x&-x)
60 #define MP make_pair
61 #define pb push_back
62 #define pt putchar
63 #define yx_queue priority_queue
64 #define lson(pos) (pos<<1)
65 #define rson(pos) (pos<<1|1)
66 #define y1 code_by_Rand0w
67 #define yn A_muban_for_ACM
68 #define j1 it_is just_an_eastegg
69 #define lr hope_you_will_be_happy_to_see_this
70 #define int long long
71 #define rep(i, a, n) for (register int i = a; i <= n;
       ++i)
72 #define per(i, a, n) for (register int i = n; i >= a;
       --i)
73 const ll llinf = 4223372036854775851;
74 const ll mod = (0 ? 1000000007 : 998244353);
75 ll pow(ll a,ll b,ll md=mod) {ll res=1;a%=md; assert(b
       >=0); for(;b;b>>=1){if(b&1)res=mul(res,a,md);a=
       mul(a,a,md);}return res;}
76 const ll mod2 = 999998639;
77 const int m1 = 998244353;
78 const int m2 = 1000001011;
79 const int pr=233;
80 const double eps = 1e-7;
81 const int maxm= 1;
82 const int maxn = 510000;
83 void work()
84 {
85
86 }
87 signed main()
88 {
89    #ifndef ONLINE_JUDGE
90    //freopen("in.txt","r",stdin);
91    //freopen("out.txt","w",stdout);
92 #endif
93    //std::ios::sync_with_stdio(false);
94    //cin.tie(NULL);
95    int t = 1;
96    //cin>>t;
97    for(int i=1;i<=t;i++){
98        //cout<<"Case #"<<i<<":"<<endll;
99        work();
100   }
101   return 0;
102 }
```

1.2　头文件 (REXWind)

```cpp
#include<iostream>
#include<cstring>
#include<cstdio>
#include<algorithm>
#include<vector>
#include<map>
#include<queue>
#include<cmath>
using namespace std;

template<class T>inline void read(T &x){x=0;char o,f
    =1;while(o=getchar(),o<48)if(o==45)f=-f;do x=(x
    <<3)+(x<<1)+(o^48);while(o=getchar(),o>47);x*=f;}
int cansel_sync=(ios::sync_with_stdio(0),cin.tie(0)
    ,0);
#define ll long long
#define ull unsigned long long
#define rep(i,a,b) for(int i=(a);i<=(b);i++)
#define repb(i,a,b) for(int i=(a);i>=b;i--)
#define mkp make_pair
#define ft first
#define sd second
#define log(x) (31-__builtin_clz(x))
#define INF 0x3f3f3f3f
typedef pair<int,int> pii;
typedef pair<ll,ll> pll;
ll gcd(ll a,ll b){ while(b^=a^=b^=a%=b); return a; }
//#define INF 0x7fffffff

void solve(){

}

int main(){
    int z;
    cin>>z;
    while(z--) solve();
}
```

1.3　头文件 (Dallby)

```cpp
#include<bits/stdc++.h>
cout<<"hello<<endl;
```

2　数论

2.1　欧拉筛

$O(n)$ 筛素数

```cpp
int primes[maxn+5],tail;
bool is_prime[maxn+5];
void euler(){
  is_prime[1] = 1;
  for (int i = 2; i < maxn; i++)
  {
    if (!is_prime[i])
    primes[++tail]=i;
    for (int j = 0; j < primes.size() && i * primes[
        j] < maxn; j++)
    {
```

```cpp
      is_prime[i * primes[j]] = 1;
      if ((i % primes[j]) == 0)
        break;
    }
  }
}
```

2.2　Exgcd

求出 $ax + by = gcd(a,b)$ 的一组可行解 $O(logn)$

```cpp
void Exgcd(ll a,ll b,ll &d,ll &x,ll &y){
    if(!b){d=a;x=1;y=0;}
    else{Exgcd(b,a%b,d,y,x);y-=x*(a/b);}
}
```

2.3　Excrt 扩展中国剩余定理

求解同余方程组 $\begin{cases} x \% b_1 \equiv a_1 \\ x \% b_2 \equiv a_2 \\ \qquad \vdots \\ x \% b_n \equiv a_n \end{cases}$

```cpp
int excrt(int a[],int b[],int n){
    int lc=1;
    for(int i=1;i<=n;i++)
        lc=lcm(lc,a[i]);
    for(int i=1;i<n;i++){
        int p,q,g;
        g=exgcd(a[i],a[i+1],p,q);
        int k=(b[i+1]-b[i])/g;
        q=-q;p*=k;q*=k;
        b[i+1]=a[i]*p%lc+b[i];
        b[i+1]%=lc;
        a[i+1]=lcm(a[i],a[i+1]);
    }
    return (b[n]%lc+lc)%lc;
}
```

2.4　线性求逆元

```cpp
void init(int p){
    inv[1] = 1;
    for(int i=2;i<=n;i++){
        inv[i] = (ll)(p-p/i)*inv[p%i]%p;
    }
}
```

2.5　组合数

预处理阶乘，并通过逆元实现相除

```cpp
ll jc[MAXN];
ll qpow(ll d,ll c){//快速幂
    ll res = 1;
    while(c){
        if(c&1) res=res*d%med;
        d=d*d%med;c>>=1;
    }return res;
}
inline ll niyuan(ll x){return qpow(x,med-2);}
```

```
10  void initjc(){//初始化阶乘
11      jc[0] = 1;
12      rep(i,1,MAXN-1) jc[i] = jc[i-1]*i%med;
13  }
14  inline int C(int n,int m){//n是下面的
15      if(n<m) return 0;
16      return jc[n]*niyuan(jc[n-m])%med*niyuan(jc[m])%med
            ;
17  }
18  int main(){
19      initjc();
20      int n,m;
21      while(cin>>n>>m) cout<<C(n,m)<<endl;
22  }
```

2.6   矩阵快速幂

```
1   struct Matrix{
2       ll a[MAXN][MAXN];
3
4       Matrix(ll x=0){//感觉是特别好的初始化,从hjt那里学(抄
            )来的
5           for(int i=0;i<n;i++){
6               for(int j=0;j<n;j++){
7                   a[i][j]=x*(i==j);//这句特简洁
8               }
9           }
10      }
11
12      Matrix operator *(const Matrix &b)const{//通过重载
            运算符实现矩阵乘法
13          Matrix res(0);
14          for(int i=0;i<n;i++){
15              for(int j=0;j<n;j++){
16                  for(int k=0;k<n;k++){
17                      ll &ma = res.a[i][j];
18                      ma = (ma+a[i][k]*b.a[k][j])%mod;
19                  }
20              }
21          }
22          return res;
23      }
24  };
25
26  Matrix qpow(Matrix d,ll m){//底数和幂次数
27      Matrix res(1);//构造E单位矩阵
28      while(m){
29          if(m&1){
30              m--;//其实这句是可以不要的
31              res=res*d;
32          }
33          d=d*d;
34          m>>=1;
35      }
36      return res;
37  }
```

2.7   高斯消元

```
1   待补充
```

2.8   三点求圆心

```
1   struct point{
2       double x;
3       double y;
4   };
5
6   point cal(point a,point b,point c){
7       double x1 = a.x;double y1 = a.y;
8       double x2 = b.x;double y2 = b.y;
9       double x3 = c.x; double y3 = c.y;
10      double a1 = 2*(x2-x1); double a2 = 2*(x3-x2);
11      double b1 = 2*(y2-y1); double b2 = 2*(y3-y2);
12      double c1 = x2*x2 + y2*y2 - x1*x1 - y1*y1;
13      double c2 = x3*x3 + y3*y3 - x2*x2 - y2*y2;
14      double rx = (c1*b2-c2*b1)/(a1*b2-a2*b1);
15      double ry = (c2*a1-c1*a2)/(a1*b2-a2*b1);
16      return point{rx,ry};
17  }
```