

Berlekamp-Massey 算法

作者: [MessiahJK](#)

原文链接: <https://ld246.com/article/1538138129410>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

黑科技算法

线性递推算法, 一定程度上可以代替矩阵快速幂

wiki: https://en.wikipedia.org/wiki/Berlekamp%E2%80%93Massey_algorithm

杜教的BM模板

```
#include <bits/stdc++.h>

using namespace std;
#define rep(i,a,n) for (long long i=a;i<n;i++)
#define per(i,a,n) for (long long i=n-1;i>=a;i--)
#define pb push_back
#define mp make_pair
#define all(x) (x).begin(),(x).end()
#define fi first
#define se second
#define SZ(x) ((long long)(x).size())
typedef vector<long long> VI;
typedef long long ll;
typedef pair<long long,long long> PII;
const ll mod=1e9+7;
ll powmod(ll a,ll b) {ll res=1;a%=mod; assert(b>=0); for(;b>=>=1){if(b&1)res=res*a%mod;a=
*a%mod;}return res;}
// head

long long _n;
namespace linear_seq
{
    const long long N=10010;
    ll res[N],base[N],_c[N],_md[N];

    vector<long long> Md;
    void mul(ll *a,ll *b,long long k)
    {
        rep(i,0,k+k) _c[i]=0;
        rep(i,0,k) if (a[i]) rep(j,0,k)
            _c[i+j]=( _c[i+j]+a[i]*b[j])%mod;
        for (long long i=k+k-1;i>=k;i--) if ( _c[i])
            rep(j,0,SZ(Md)) _c[i-k+Md[j]]=( _c[i-k+Md[j]]- _c[i]*_md[Md[j]])%mod;
        rep(i,0,k) a[i]=_c[i];
    }
    long long solve(ll n,VI a,VI b)
    { // a 系数 b 初值 b[n+1]=a[0]*b[n]+...
        printf("%d\n",SZ(b));
        ll ans=0,pnt=0;
        long long k=SZ(a);
        assert(SZ(a)==SZ(b));
        rep(i,0,k) _md[k-1-i]=-a[i];_md[k]=1;
        Md.clear();
        rep(i,0,k) if ( _md[i]!=0) Md.push_back(i);
        rep(i,0,k) res[i]=base[i]=0;
        res[0]=1;
        while ((1ll<<pnt)<=n) pnt++;
    }
}
```

```

for (long long p=pnt;p>=0;p--)
{
    mul(res,res,k);
    if ((n>>p)&1)
    {
        for (long long i=k-1;i>=0;i--) res[i+1]=res[i];res[0]=0;
        rep(j,0,SZ(Md)) res[Md[j]]=(res[Md[j]]-res[k]*_md[Md[j]])%mod;
    }
}
rep(i,0,k) ans=(ans+res[i]*b[i])%mod;
if (ans<0) ans+=mod;
return ans;
}
VI BM(VI s)
{
    VI C(1,1),B(1,1);
    long long L=0,m=1,b=1;
    rep(n,0,SZ(s))
    {
        ll d=0;
        rep(i,0,L+1) d=(d+(ll)C[i]*s[n-i])%mod;
        if (d==0) ++m;
        else if (2*L<=n)
        {
            VI T=C;
            ll c=mod-d*powmod(b,mod-2)%mod;
            while (SZ(C)<SZ(B)+m) C.pb(0);
            rep(i,0,SZ(B)) C[i+m]=(C[i+m]+c*B[i])%mod;
            L=n+1-L; B=T; b=d; m=1;
        }
        else
        {
            ll c=mod-d*powmod(b,mod-2)%mod;
            while (SZ(C)<SZ(B)+m) C.pb(0);
            rep(i,0,SZ(B)) C[i+m]=(C[i+m]+c*B[i])%mod;
            ++m;
        }
    }
    return C;
}
long long gao(VI a,ll n)
{
    VI c=BM(a);
    c.erase(c.begin());
    rep(i,0,SZ(c)) c[i]=(mod-c[i])%mod;
    return solve(n,c,VI(a.begin(),a.begin()+SZ(c)));
}
};

int main()
{
    while(~scanf("%l64d", &n))
    { printf("%l64d\n",linear_seq::gao(VI{1,5,11,36,95,281,781,2245,6336,18061, 51205},n-1));
    }
}

```

}