

牛客练习赛83

[比赛链接](#)

D-数列递推

题意

给定 n, f_0 求 $f_{1 \sim n}$ 其中

$$f_i = \sum_{j=1}^i f_{i \bmod j}$$

题解

设 $i = kb + r$ 考虑整数分块枚举 k 设 $t = i \bmod k$ 此时有 $r = t, t+k, t+2k, \dots, i-k$

当 $k \leq \sqrt{n}$ 时，如果能提前维护 $f_t, f_{t+2k}, f_{t+3k}, \dots$ 的前缀和，就可以 $O(1)$ 计算贡献。

当 $k \geq \sqrt{n}$ 时，显然 b, r 唯一，直接计算贡献即可。于是整数分块部分的时间复杂度为 $O(\sqrt{i})$

对每个 i 枚举 $k \leq \sqrt{n}$ 更新 $f_t, f_{t+2k}, f_{t+3k}, \dots$ 的前缀和的时间复杂度为 $O(\sqrt{n})$ 于是总时间复杂度 $O(n\sqrt{n})$

```
const int MAXN=1e5+5,MAXM=405,Mod=998244353;
int f[MAXN];
int s[MAXM][MAXN];
int main()
{
    int n=read_int(),v0=read_int(),m=sqrt(n)+1;
    f[0]=1;
    _for(i,0,m)
        s[i][0]=1;
    _rep(i,1,n){
        int lef=1,rig=0;
        while(lef<=i){
            rig=i/(i/lef);
            int k=i/lef;
            if(k<m)
                f[i]=(f[i]+s[k][i-lef*k])%Mod;
            else
                f[i]=(f[i]+f[i-lef*k])%Mod;
            lef=rig+1;
        }
        _for(k,1,m){
```

```
        if(i>=k)
            s[k][i]=(s[k][i-k]+f[i])%Mod;
        else
            s[k][i]=f[i];
    }
}
_rep(i,1,n){
    f[i]=(1LL*f[i]*v0%Mod+Mod)%Mod;
    space(f[i]);
}
return 0;
}
```

From:
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:jxm2001:contest:%E7%89%9B%E5%AE%A2%E7%BB%83%E4%B9%A0%E8%B5%9B83&rev=1621650815

Last update: 2021/05/22 10:33