生成函数理论 5

基本应用

例 1

P2000 拯救世界

依照题意可列出 10 个条件的生成函数 $f_i(x)$，将它们相乘再化简即可。

- $f_1(x)=(1+x^6+x^{12}+\cdots)=\frac{1}{1-x^6}$
- $f_2(x)=(1+x+x^2+\cdots+x^9)=\frac{1-x^{10}}{1-x}$
- $f_3(x)=(1+x+x^2+\cdots+x^5)=\frac{1-x^6}{1-x}$
- $f_4(x)=(1+x^4+x^8+\cdots)=\frac{1}{1-x^4}$
- $f_5(x)=(1+x+x^2+\cdots+x^7)=\frac{1-x^8}{1-x}$
- $f_6(x)=(1+x^2+x^4+\cdots)=\frac{1}{1-x^2}$
- $f_7(x)=(1+x)=\frac{1-x^2}{1-x}$
- $f_8(x)=(1+x^8+x^{16}+\cdots)=\frac{1}{1-x^8}$
- $f_9(x)=(1+x^{10}+x^{20}+\cdots)=\frac{1}{1-x^{10}}$
- $f_{10}(x)=(1+x+x^2+x^3)=\frac{1-x^4}{1-x}$

$\displaystyle f(x)=\prod_{i=1}^{10}f_i(x)=\frac{1}{(1-x)^5}=\sum_{n=0}^{\infty}\binom{n+4}{n}x^n$

$\displaystyle ans_n=\binom{n+4}{4}=\frac{(n+4)(n+3)(n+2)(n+1)}{24}$

由于 $10^{100000}\le n<10^{100001}$，需要用高精度，正解是用 NTT，用 Python 会 TLE，但是竟然可以用 Ruby AC。

Ruby 代码：

```ruby
n=Integer(gets)
puts (n+4)*(n+3)*(n+2)*(n+1)/24
```

C++ 代码：

```cpp
#include<bits/stdc++.h>
using namespace std;
const int N=5e6+5,P=998244353,G=3,Gi=332748118;
typedef long long ll;
int rev[N];
char s[N];
ll n1[N],n2[N],n3[N],n4[N];
ll fastpow(ll x,ll y){
```

```
    ll ret=1;
    for(;y;y>>=1,x=x*x%P)
        if(y&1) ret=ret*x%P;
    return ret;
}
void NTT(ll *y,int len,int on){
    for(int i=0;i<len;i++)
        if(i<rev[i])
            swap(y[i],y[rev[i]]);
    for(int mid=1;mid<len;mid<<=1){
        ll wn=fastpow(on==1?G:Gi,(P-1)/(mid<<1));
        for(int j=0;j<len;j+=(mid<<1)){
            ll w=1;
            for(int k=0;k<mid;k++,w=w*wn%P){
                ll u=y[j+k],t=w*y[j+k+mid]%P;
                y[j+k]=(u+t)%P;
                y[j+k+mid]=(u-t+P)%P;
            }
        }
    }
}
ll c[N];
void mul(ll *a,ll *b,int &n,int &m){
    for(int i=0;i<=n;i++) a[i]%=P;
    for(int i=0;i<=m;i++) b[i]%=P;
    int len=1,l=0;
    while(len<=n+m) len<<=1,l++;
    for(int i=0;i<len;i++) rev[i]=(rev[i>>1]>>1)|((i&1)<<(l-1));
    NTT(a,len,1);
    NTT(b,len,1);
    for(int i=0;i<len;i++) a[i]=a[i]*b[i]%P;
    NTT(a,len,-1);
    ll inv=fastpow(len,P-2);
    memset(c,0,sizeof(c));
    for(int i=0;i<=n+m;i++){
        a[i]=a[i]*inv%P;
        c[i]+=a[i];
        c[i+1]+=c[i]/10;
        c[i]%=10;
    }
    if(c[n+m+1]) n++;
    for(int i=0;i<=n+m;i++) a[i]=c[i];
    n=n+m;
}
ll ans[N];
int main(){
    scanf("%s",s);
    int n=strlen(s);
    n--;
    int len1=n,len2=n,len3=n,len4=n;
    for(int i=0;i<=n;i++) n1[i]=n2[i]=n3[i]=n4[i]=s[n-i]-'0';
```

```
    n1[0]+=1;
    n2[0]+=2;
    n3[0]+=3;
    n4[0]+=4;
    for(int i=0;i<=n;i++){
        n1[i+1]+=n1[i]/10;
        n1[i]%=10;
        n2[i+1]+=n2[i]/10;
        n2[i]%=10;
        n3[i+1]+=n3[i]/10;
        n3[i]%=10;
        n4[i+1]+=n4[i]/10;
        n4[i]%=10;
    }
    if(n1[n+1]) len1++;
    if(n2[n+1]) len2++;
    if(n3[n+1]) len3++;
    if(n4[n+1]) len4++;
    mul(n1,n2,len1,len2);
    mul(n1,n3,len1,len3);
    mul(n1,n4,len1,len4);

    ll p=0;
    int lenans=0;
    for(int i=len1;i>=0;i--){
        p=p*10+n1[i];
        ans[i]=p/24;
        p%=24;
        if(ans[i]&&!lenans) lenans=i;
    }
    for(int i=lenans;i>=0;i--)
        printf("%lld",ans[i]);
    return 0;
}
```