

算法设计与分析 C6-C

21377206 阮阳栋

题目描述

本题的MIPS指令集中立即数均为八进制，并且系统调用会以八进制输出数字，即给定两个八进制非负整数 a, b ，输出相乘的结果。

题目分析

分析方法与B题类似， a, b 这两个八进制数可以表示为如下形式 ($x = 8$)：

$$a = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

$$b = b_0 + b_1x + b_2x^2 + \dots + b_mx^m$$

这就把此题转化为了多项式乘法问题，即求解下面多项式的系数：

$$a \times b = c_0 + c_1x + c_2x^2 + \dots + c_{m+n}x^{m+n}$$

题目求解

由于 a, b 的取值范围为 $[0, 8^{10^5})$ ，不能直接遍历进行相乘，需要使用FFT算法。

```
void fft(complex *A, int type){
    for(int i=0;i<limit;i++)
        if(i<r[i]) swap(A[i],A[r[i]]);
    for(int mid=1;mid<limit;mid<=1){
        complex wn(cos(Pi/mid), type*sin(Pi/mid));
        for(int R=mid<<1,j=0;j<limit;j+=R){
            complex w(1,0);
            for(int k=0;k<mid;k++,w=w*wn){
                complex x=A[j+k],y=w*A[j+mid+k];
                //蝴蝶变换
                A[j+k]=x+y, A[j+mid+k]=x-y;
            }
        }
    }
}
```

在完成FFT后，需要对每一个系数进行进位处理：

```

for (int i=HH+n+m;i>=0;i--){
    int u;
    if (i>=HH) u = (ll)(a[i-HH].x/limit+0.5);
    else u = 0;
    if (c0+u < 8) result[i] = c0+u, c0 = 0;
    else result[i] = (c0+u)%8, c0 = (c0+u)/8;
}

```

可以发现把上面代码的 8 都改成 10 就可以通过B题。

时间复杂度

FFT的时间复杂度为 $O(n \log n)$ ，后续处理为 $O(n)$ ，总时间复杂度为 $O(n \log n)$ 。

核心代码

使用FFT求解的大体思路：

```

fftPre(), fft(a, 1), fft(b, 1); //fft
for(int i=0;i<=limit;i++) a[i] = a[i]*b[i]; //每点相乘
fft(a,-1); //fft逆变换

int c0 = 0; //进位
for (int i=HH+n+m;i>=0;i--){
    int u; //相乘后的多项式系数
    if (i>=HH) u = (ll)(a[i-HH].x/limit+0.5);
    else u = 0;
    if (c0+u < 8) result[i] = c0+u, c0 = 0;
    else result[i] = (c0+u)%8, c0 = (c0+u)/8;
}

```