

Atcoder Rugular Contest 122

[比赛链接](#)

C - Calculator

题意

给定 x, y 初值均为 0 ，接下来给定 4 种操作：

1. $x \leftarrow x+1$
2. $y \leftarrow y+1$
3. $x \leftarrow x+y$
4. $y \leftarrow x+y$

要求在 130 步操作内将 x 变为 $N (N \leq 10^{18})$

赛时解法

不妨先规定一个 y 的最终值，然后利用操作 $3, 4$ 对 x, y 进行更相减损术，当 x, y 其中一个为 0 时再利用操作 $1, 2$ 暴力处理。

最后逆序输出即可。操作次数等于更相减损术次数加上 $\text{gcd}(x, y)$ 不难发现斐波那契数列是最理想的情况，但 N 不一定是斐波那契数。

不妨强制认为 N 是斐波那契数，于是根据斐波那契数通项公式不妨猜想 y 的最终值在 $\frac{\{2N\} \cdot \sqrt{5} + 1}{5}$ 附近。

将 $\frac{\{2N\} \cdot \sqrt{5} + 1}{5} - 20$ 到 $\frac{\{2N\} \cdot \sqrt{5} + 1}{5} + 20$ 都代入尝试即可。

```
LL cal(LL n, LL p){
    int pos=3;
    LL ans=0;
    while(p>0){
        LL t=n;
        while(t>=p){
            t-=p;
            ans++;
        }
        if(pos==3) pos=4;
        else pos=3;
        n=p;
        p=t;
    }
    return ans+n;
}
```

```
int main()
{
    LL n=read_LL();
    LL v=n*2/(sqrt(5)+1);
    for(LL i=max(v-20,0LL);i<=min(v+20,n);i++){
        if(cal(n,i)<125){
            LL p=i;
            stack<int> s;
            int pos=3;
            while(p>0){
                LL t=n;
                while(t>=p){
                    t-=p;
                    s.push(pos);
                }
                if(pos==3)pos=4;
                else pos=3;
                n=p;
                p=t;
            }
            enter(s.size()+n);
            LL tn=0,tp=0;
            _for(i,0,n){
                if(pos==3){
                    tn++;
                }
                else
                    tp++;
                enter(pos-2);
            }
            while(!s.empty()){
                enter(s.top());
                if(s.top()==3)
                    tn=tn+tp;
                else
                    tp=tn+tp;
                s.pop();
            }
            return 0;
        }
    }
    return 0;
}
```

正解

假定操作序列为 $4,3,4,3,4,3\cdots$ 共操作 S 次，且最后一次操作为 3 。

接下来考虑在该操作序列中插入 $1,2$ 操作，定义 $F(0)=F(1)=1, F(n)=F(n-1)+F(n-2)$

不妨规定仅在操作 \$3\$ 后面或者最开始插入操作 \$1\$，操作 \$4\$ 后面插入操作 \$2\$。

不难发现在第 \$i(0 \leq i \leq S)\$ 次操作后插入一个操作最后对 \$x\$ 的贡献为 \$F(S-i)\$

于是问题转化为将 \$N\$ 分解为若干斐波那契数。取 \$S = \max\{F(S) \leq 10^{18}\} = 86\$

最后贪心分解 \$N\$ 即可。显然 \$F(i), F(i-1)\$ 不可能同时存在与 \$N\$ 的分解，否则可以用 \$F(i+1)\$ 替代，于是最大操作数为 \$\lceil \frac{(S+1)}{2} \rceil = 44\$

于是总操作数不超过 \$130\$ 次。

From:

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

Permanent link:

https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:jxm2001:contest:arc_122&rev=1623833578

Last update: **2021/06/16 16:52**