

# LAB\_03 实验二优化 (BETTER ANGEL)

## 阅读代码

### 代码解读

从仓库得到代码如下，经过阅读后进行注释。

```
        ADD R1, R1, #1
        ADD R2, R2, #2
        ADD R3, R3, #4; 依次存入f1, f2, f3的值
        LD  R4, MOD//存入1023（取模）

//进入循环
DECIDE   ADD R0, R0, #-1; R0存入n的值，判断是否完成循环
        BRz OVER//R0为0代表循环结束（进行了R0次运算）

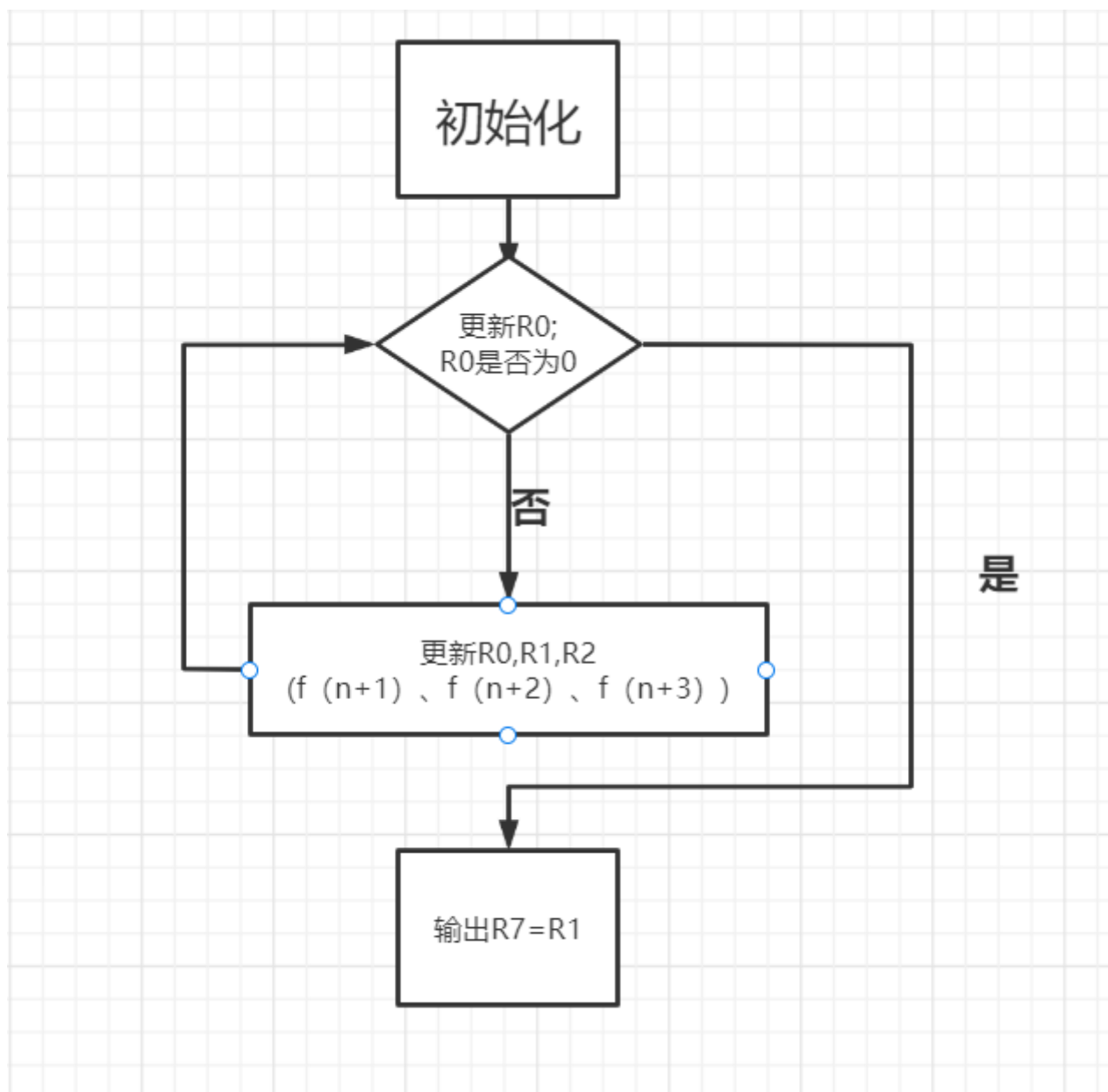
CALC     ADD R7, R3, #0//将R3存入R7，之后n值更新
        ADD R3, R3, R1
        ADD R3, R3, R1; R3=2*f(n-3)
        AND R3, R3, R4; R3= (2*f(n-3)+f(n-1)) %1024
        ADD R1, R2, #0
        ADD R2, R7, #0; 更新R1, R2的值
        BRnzp DECIDE

OVER     ADD R7, R1, #0; 存入R1的值
        ;实际上R1, R2, R3分别存储的是f(n), f(n+1), f(n+2)
        TRAP x25//HALT

MOD      .FILL #1023
FA       .FILL #930
FB       .FILL #18
FC       .FILL #710
FD       .FILL #370
```

## 算法思路

画出流程图如下



## 判断学号

该同学学号4部分分别为930，18，710，370，用投机取巧的办法可以知道第一部分数字大概率为20，之后按照学校学号编码中筛选得到18对应的院系是“06”信院。锁定范围后，在群聊中判断得到后四位为1338。故得出学号应为PB20061338。

当然这显然有更正统的判断方法。我们可以通过一个简单的C程序进行查找。

```
#include <stdio.h>
#include<stdlib.h>
```

```

int f(int n0, int n1, int n2, int n) {
    int fn;
    fn=(2*n0+n2)%1024;
    if (n==0)
        return 1;
    else if (n == 1)
        return 1;
    else if (n == 2)
        return 2;
    else if(n==3)
        return fn;
    else if(n>3)
        return f(n1, n2, fn, n-1);
    else
        return 0;
} //数列函数

int main() {
    int n;

    scanf("%d", &n); //输入对应的值
    for ( int i = 200; i < 1000; i++)
    {
        for(int j = 0; j < n; j++)

            if(f(1, 1, 2, i)==f(1, 1, 2, i+j)&&f(1, 1, 2, i+1)==f(1, 1, 2, i+j+1)&&f(1, 1, 2, i+2)==f(1, 1, 2, i+j+2))
            {
                printf("%d\n", j);
            }
    }
    system("pause");
    return 0;
}

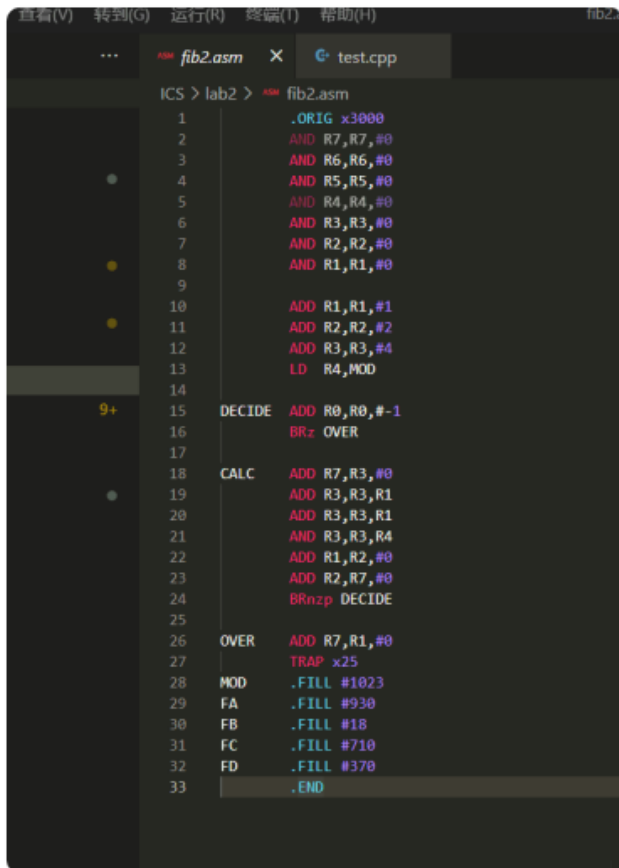
```

最后得出答案仍是PB20061338。

找到该目标同学进行验证，得到肯定回复。

打开 打开文件夹 重新下载 转发

图片可以吗?



```
ICS > lab2 > fib2.asm
1      .ORIG x3000
2      AND R7,R7,#0
3      AND R6,R6,#0
4      AND R5,R5,#0
5      AND R4,R4,#0
6      AND R3,R3,#0
7      AND R2,R2,#0
8      AND R1,R1,#0
9
10     ADD R1,R1,#1
11     ADD R2,R2,#2
12     ADD R3,R3,#4
13     LD  R4,MOD
14
15     DECIDE ADD R0,R0,#-1
16           BRz OVER
17
18     CALC  ADD R7,R3,#0
19           ADD R3,R3,R1
20           ADD R3,R3,R1
21           AND R3,R3,R4
22           ADD R1,R2,#0
23           ADD R2,R7,#0
24           BRnzp DECIDE
25
26     OVER  ADD R7,R1,#0
27           TRAP x25
28     MOD   .FILL #1023
29     FA    .FILL #930
30     FB    .FILL #18
31     FC    .FILL #710
32     FD    .FILL #370
33     .END
```

好的

## 代码最佳化

### 最优解

通过比对数据不难发现，在f(20)之后，代码将会以128为周期进行循环，而且助教指定了测试数据，因此，我们可以进行打表直接读取。

```
LEA R1,ORG;
ADD R1,R1,R0;
LDR R7,R1,#0;
HALT
ORG .FILL #1
;之后写入f1到f16384的值
```

这种方法只需要3个运算周期，属于最极限解法，但是占用了较大空间，而且对超出16384的n都不适用，所以需要一种更有效的解法。

## 普适最优解

既然确定这些数据是具有周期性的，那么只需要148个数据进行取模运算即可，所以可以得出以下算法。

```
. ORG x3000
LD R1, a ;a是-20
ADD R1, R0, R1;判断n是否大于20
BRp NOR;是，进入循环判断中取模，不是进入非循环语句中进行判断
LEA R2 Nloop;R2存入f0的地址
ADD R2, R0, R2
LDR R7 R2 #0;存入R2中地址对应的值
BRnzp END
NOR LD R3 mod;mod是127，用于取模操作（128为一个周期）
AND R3 R1 R3;取模
LEA R4 Loop;载入循环序列的首位地址（f20）
ADD R4 R4 R3
LDR R7 R4 #0;存入R4中地址对应值
END HALT
a .FILL #-20
mod .FILL #127
Nloop .FILL #1
... (其余f1至f19)
Loop .FILL #930
... (f21到f148)
.end
```

经过简单抽样，此算法的周期在6.7左右。