Lab Report - 03

PB20111686 Ruixuan Huang

Lab Goal

1. Translate the machine code in `foo.txt` into assembly code and store it in `translate.txt`.
2. Guess the owner of the program by the last 4 lines of the program and write down the owner's student id in `id.txt`.
3. Optimize program code in `foo.txt`. The optimization direction is its performance.

Task 1: Read

Here's the content of my `foo.txt` and I translate the machine code into assembly language code, which is presented in form of comments.

```
0011000000000000     ; .Orig x3000
0001 111 111 1 00010 ; x3000, ADD R7, R7, #2
0001 010 010 1 00001 ; x3001, ADD R2, R2, #1
0001 001 001 1 00001 ; x3002, ADD R1, R1, #1
0010 011 000010001   ; x3003, LD, R3, x3015
0001 000 000 1 11110 ; x3004, ADD, R0, R0, #-2
0000 100 000001001   ; x3005, BRn, x300F
0000 010 000001001   ; x3006, BRz, x3010
0001 000 000 1 11111 ; x3007, ADD, R0, R0, #-1
0000 100 000000111   ; x3008, BRn, x3010
0001 100 001 000 001 ; x3009, ADD, R4, R1, R1
0001 001 010 1 00000 ; x300A, ADD, R1, R2, #0
0001 010 111 1 00000 ; x300B, ADD, R2, R7, #0
0001 111 111 000 100 ; x300C, ADD, R7, R7, R4
0101 111 111 000 011 ; x300D, AND, R7, R7, R3
0000 111 111111000   ; x300E, BRnzp, x3007
0001 111 111 1 11111 ; x300F, ADD, R7, R7, #-1
1111000000100101     ; HALT
0000 0010 1101 0010  ; x02D2 #722 No.18
0000 0000 1111 0110  ; x00F6 #246 No.11
0000 0010 1110 0110  ; x02E6 #742 No.17
0000 0010 0010 0110  ; x0226 #550 No.41
0000 0011 1111 1111  ; x03FF , x3015
```

Note that instructions `x3015, x300F, x3010, x3007` may be the destination of other instructions. We should give them labels. The final assembly code is as follows.

```
       .ORIG x3000
       ADD   R7, R7, #2
       ADD   R2, R2, #1
       ADD   R1, R1, #1
       LD    R3, MOD
       ADD   R0, R0, #-2
       BRn   DF
       BRz   D10
D7     ADD   R0, R0, #-1
       BRn   D10
       ADD   R4, R1, R1
       ADD   R1, R2, #0
       ADD   R2, R7, #0
       ADD   R7, R7, R4
       AND   R7, R7, R3
       BRnzp D7
```

```
DF      ADD   R7, R7, #-1
D10     HALT
fa      .FILL x02D2  ;#722 No.18
fb      .FILL x00F6  ;#246 No.11
fc      .FILL x02E6  ;#742 No.17
fd      .FILL x0226  ;#550 No.41
MOD     .FILL x03FF  ;x3015
        .END
```

Task 2: Guess

I use C++ programming to generate the list of the first 100+ items of this sequence, which is presented in the task 3 part of this lab report. Refer to the list, and I can give the student ID of the person who is the author of the code.

```
18111741
```

I refer the student ID to the `USTC/young database`, and find the student's name. And I refer his name to the ics-Feishu group, verifying that he do is the student of our class.

Task 3: Optimize

Here's a conclusion in maths : Any homogeneous linear integer recurrence sequence is modulo periodic.

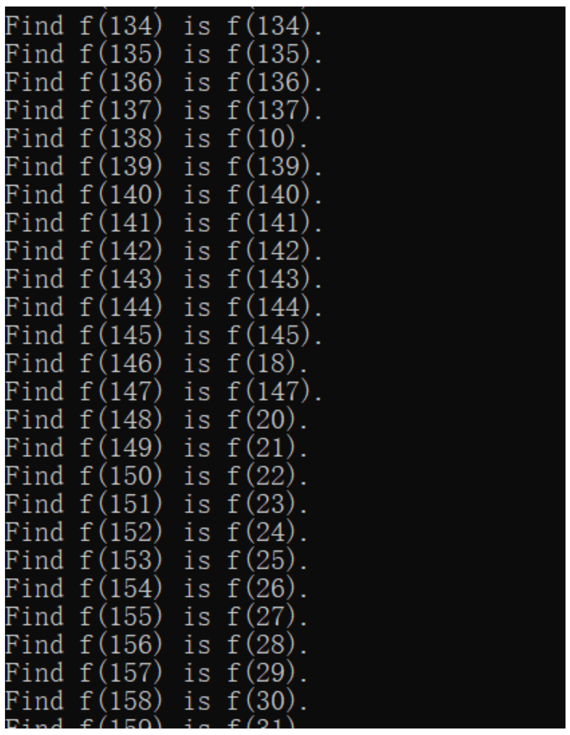
> ***Proof*** *: We consider a m-order homogeneous linear recursive integer sequence with constant coefficients, whose definiation is*
>
> $$g(n) = k_1 g(n-1) + \ldots + k_m g(n-m)$$
>
> *Here* $n \in \mathbb{N}, m \in \mathbb{N}^*, k_1, \ldots, k_m$ *are constant unegative integer.*
>
> *What we concern about is another sequence* $f(n)$*, whose defination is*
>
> $$f(n) = g(n) \bmod p, p \in \mathbb{N}^*$$
>
> *Consider m-ary ordered array*
>
> $$(f(n-m), f(n-m+1), \ldots, f(n-1)), f(i) \in (0, 1, 2, \ldots, p-1), i \in \mathbb{N}^*$$
>
> *There're at most* p^m *such arrays. If we considered infinite numbers of items, there must be two arrays that are the same. Here comes the periodicity. Although the sequence isn't being periodic from its first item.*

I use C++ programming to find the period of $f(n)$. My idea is using vector to store newly caculated items. If $f(n)$ appeared in the vector, say $f(k), k < n$, then find if $f(n+1) = f(k+1), f(n+2) = f(k+2)$. If so, for this lab, we can declare the periodicity is found.

```
#define int short

int find(vector<int>* v, int r) {
    int size = v->size();
    for (int i = 0; i < size; i++) {
        if (v->at(i) == r) {
            return i;
        }
    }
    return -1;
}
int main() {
    vector<short> ans;
    for (short i = 1; i < 200; i++) {
        int l = lab2(i);
        ans.push_back(l);
    }
    for (short i = 1; i < 200; i++) {
        cout << "Find f(" << i << ") is f(" << find(&ans, ans[i - 1]) + 1 << ")." << endl;
    }
}
```

Observing the first 200 items we can get the point.



```
Find f(134) is f(134).
Find f(135) is f(135).
Find f(136) is f(136).
Find f(137) is f(137).
Find f(138) is f(10).
Find f(139) is f(139).
Find f(140) is f(140).
Find f(141) is f(141).
Find f(142) is f(142).
Find f(143) is f(143).
Find f(144) is f(144).
Find f(145) is f(145).
Find f(146) is f(18).
Find f(147) is f(147).
Find f(148) is f(20).
Find f(149) is f(21).
Find f(150) is f(22).
Find f(151) is f(23).
Find f(152) is f(24).
Find f(153) is f(25).
Find f(154) is f(26).
Find f(155) is f(27).
Find f(156) is f(28).
Find f(157) is f(29).
Find f(158) is f(30).
Find f(159) is f(31).
```

To verify, I use another program.

```
int main() {
    for (short i = 1; i < 400; i++) {
        if (lab2(i) == lab2(i + 128))cout << "1";
        else cout << "0";
    }
}
```

The result is:

```
00000000010000000101111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111
```

So we can store the first 19 items and next 128 items in a table and using the table to optimize the program's performance. My solutions is as follows.

```
      .ORIG x3000
      LEA   R2, f19
      ADD   R0, R0, #-16
      ADD   R0, R0, #-3  ; R0 = R0 - 19
      BRnz  NOTP
      LD    R1, PER
      AND   R0, R0, R1
NOTP  ADD   R2, R2, R0
      LDR   R7, R2, #0
      HALT

PER   .FILL #127
PMAX  .FILL #147
f1    .FILL #1
f2    .FILL #2
f3    .FILL #4
f4    .FILL #6
f5    .FILL #10
f6    .FILL #18
f7    .FILL #30
```

```
f8    .FILL #50
f9    .FILL #86
f10   .FILL #146
f11   .FILL #246
f12   .FILL #418
f13   .FILL #710
f14   .FILL #178
f15   .FILL #1014
f16   .FILL #386
f17   .FILL #742
f18   .FILL #722
f19   .FILL #470
f20   .FILL #930
f21   .FILL #326
f22   .FILL #242
f23   .FILL #54
f24   .FILL #706
f25   .FILL #166
f26   .FILL #274
f27   .FILL #662
f28   .FILL #994
f29   .FILL #518
f30   .FILL #818
f31   .FILL #758
f32   .FILL #770
f33   .FILL #358
f34   .FILL #850
f35   .FILL #342
f36   .FILL #34
f37   .FILL #710
f38   .FILL #370
f39   .FILL #438
f40   .FILL #834
f41   .FILL #550
f42   .FILL #402
f43   .FILL #22
f44   .FILL #98
f45   .FILL #902
f46   .FILL #946
f47   .FILL #118
f48   .FILL #898
f49   .FILL #742
f50   .FILL #978
f51   .FILL #726
f52   .FILL #162
f53   .FILL #70
f54   .FILL #498
f55   .FILL #822
f56   .FILL #962
f57   .FILL #934
f58   .FILL #530
f59   .FILL #406
f60   .FILL #226
f61   .FILL #262
f62   .FILL #50
f63   .FILL #502
f64   .FILL #2
f65   .FILL #102
f66   .FILL #82
f67   .FILL #86
f68   .FILL #290
f69   .FILL #454
f70   .FILL #626
f71   .FILL #182
f72   .FILL #66
f73   .FILL #294
```

```
f74   .FILL #658
f75   .FILL #790
f76   .FILL #354
f77   .FILL #646
f78   .FILL #178
f79   .FILL #886
f80   .FILL #130
f81   .FILL #486
f82   .FILL #210
f83   .FILL #470
f84   .FILL #418
f85   .FILL #838
f86   .FILL #754
f87   .FILL #566
f88   .FILL #194
f89   .FILL #678
f90   .FILL #786
f91   .FILL #150
f92   .FILL #482
f93   .FILL #6
f94   .FILL #306
f95   .FILL #246
f96   .FILL #258
f97   .FILL #870
f98   .FILL #338
f99   .FILL #854
f100  .FILL #546
f101  .FILL #198
f102  .FILL #882
f103  .FILL #950
f104  .FILL #322
f105  .FILL #38
f106  .FILL #914
f107  .FILL #534
f108  .FILL #610
f109  .FILL #390
f110  .FILL #434
f111  .FILL #630
f112  .FILL #386
f113  .FILL #230
f114  .FILL #466
f115  .FILL #214
f116  .FILL #674
f117  .FILL #582
f118  .FILL #1010
f119  .FILL #310
f120  .FILL #450
f121  .FILL #422
f122  .FILL #18
f123  .FILL #918
f124  .FILL #738
f125  .FILL #774
f126  .FILL #562
f127  .FILL #1014
f128  .FILL #514
f129  .FILL #614
f130  .FILL #594
f131  .FILL #598
f132  .FILL #802
f133  .FILL #966
f134  .FILL #114
f135  .FILL #694
f136  .FILL #578
f137  .FILL #806
f138  .FILL #146
f139  .FILL #278
```

```
f140  .FILL #866
f141  .FILL #134
f142  .FILL #690
f143  .FILL #374
f144  .FILL #642
f145  .FILL #998
f146  .FILL #722
f147  .FILL #982
      .END
```