实验报告——lab3 better angels

PB20151793 宋玮

1. 分析原程序:

```
.ORIG x3000

ADD R1, R1, #1

ADD R2, R2, #1

ADD R3, R3, #2

ADD R7, R7, R0

ADD R0, R0, #-2

BRNZ STOP

AGAIN2 ADD R7, R3, R1

ADD R7, R7, R1

LD R5, MOD

AND R7, R7, R5

ADD R1, R2, #0

ADD R2, R3, #0

ADD R3, R7, #0

ADD R0, R0, #-1

BRP AGAIN2

STOP

HALT

MOD

.FILL #1023

Fa .FILL #306

.FILL #1

Fc .FILL #1

Fc .FILL #1

Fc .FILL #1

Fc .FILL #306
```

从最后四行

Fa . FILL #930
Fb . FILL #1
Fc . FILL #1
Fd . FILL #306

#1 可以是 f(0), 也可以是 f(1), 经过对班级名单的查询,得出 b=0, c=1。

于是 a=20, b=0, c=1, d=94.

可知这位同学的学号为 PB20000194.

2. 优化过程

(1)原程序:

```
.ORIG x3000

ADD R1, R1, #1

ADD R2, R2, #1

ADD R3, R3, #2

ADD R7, R7, R0

ADD R0, R0, #-2

BRNZ STOP

AGAIN2

ADD R7, R7, R1

LD R5, MOD

AND R7, R7, R5

ADD R1, R2, #0

ADD R2, R3, #0

ADD R3, R7, #0

ADD R0, R0, #-1

BRP AGAIN2

STOP

HALT

MOD

Fill #1023

Fa .FILL #930

Fb .FILL #1

Fc .FILL #1

Fc .FILL #1

Fd .FILL #306
```

n=1 和 n=2 的情况下,程序只需执行 6 条指令。(不包括 halt)

除此之外,程序执行的指令条数都与 n 存在线性关系。而该线性系数即为循环体 AGAIN2 的指令条数。在原程序中,循环体指令条数为 9 条。因此可以得出指令执行总条数 g 与 n 的关系式。如下:

g(n) = 9(n-2) + 6

测试数据:

```
f(24)=706 指令数: 9×(24-2) +6 =204.
f(144)=642 同理得: 指令数: 1284
f(456)=66 同理得: 指令数: 4092
f(1088)=2 同理得: 指令数: 9780
f(1092)=290 同理得: 指令数: 9816
```

f(1088)=2 同理得:指令数:9780 f(1092)=290 同理得:指令数:9816 f(2096)=898 同理得:指令数:18852 f(4200)=322 同理得:指令数:37788 f(8192)=514 同理得:指令数:73716 f(12000)=258 同理得:指令数:107988 f(14000)=898 同理得:指令数:125988

平均指令条数为: 38950.8

(2)优化方案

①减少循环体指令条数。

通过观察原程序,发现有两条指令可以搬移到循环体外,不需要每次循环执行。

即 LD R5, MOD 和 AND R7, R7, R5

这两条指令的作用是对结果 mode 1024,显然可以放到循环体外,只需各执行一次。 修改完的程序如下:

```
.ORIG x3000
        ADD R1, R1, #1
         ADD R2, R2, #1
         ADD R3, R3, #2
         LD R5, MOD
         ADD R7, R7, R0
         ADD R0, R0, \#-2
         BRnz STOP
AGAIN2 ADD R7, R3, R1
         ADD R7, R7, R1
ADD R1, R2, #0
ADD R2, R3, #0
         ADD R3, R7, #0
         ADD RO, RO, \#-1
         BRp AGAIN2
         AND R7, R7, R5
STOP
         HALT
         .FILL #1023
MOD
         .FILL #930
Fa
         .FILL #1
Fb
Fc
         .FILL #1
Fd
         .FILL #306
.END
```

此时循环体的指令条数缩减至7条。

n=1 和 n=2 的情况下,程序只需执行 7 条指令。(不包括 halt)

除此之外,指令执行总条数 g 与 n 的关系式。如下:

g(n)=7(n-2)+8

根据对原程序的分析方法,同理可得在 24, 144, 456, 1088, 1092, 2096, 4200, 8192, 12000, 14000 这十个测试数据下的平均指令执行条数为 30298.4

优化比率为: 30298.4/38950.8≈77.8%

②寻找周期规律

由于在计算 f(n) 时,我们对结果进行了 $mode\ 1024$ 的操作。在测试数据时,我也发现了一些 n 值不同,但结果相同的数据。因此,我猜想,f(n) 的结果应该是一个有周期的数列。

于是, 我修改 lab2 中编写的 C 程序, 打印了 3<n<1000 的结果。

4	6 662	10 994	18 518	30 818	50 758	86 770	146 358	246 850	418 342	710 34	178 710	1014 370	386 438	742 834	722 550	470 402	930 22	326 98	242 902	54 946	706 118	166 898
2	978	726	162	70	498	822	962	934	530	406	226	262	50	502	2	102	82	86	290	454	626	182
	294	658	790	354	646	178	886	130	486	210	470	418	838	754	566	194	678	786	150	482		306
	258	870	338	854	546	198	882	950	322	38	914	534	610	390	434	630	386	230	466	214	674	582
)	310	450	422	18	918	738	774	562	1014	514	614	594	598	802	966	114	694	578	806	146	278	866
	690 710	374	642 438	998	722	982	930	326	242	54	706	166	274	662 978	994 726	518	818	758	770	358	850	342
	226	370 262	438 50	834 502	550 2	402 102	22 82	98 86	902 290	946 454	118 626	898 182	742 66	978 294	658	162 790	70 354	498 646	822 178	962 886	934 130	530 486
	470	418	838	754	566	194	678	786	150	482	6	306	246	258	870	338	854	546	198	882	950	322
	914	534	610	390	434	630	386	230	466	214	674	582	1010	310	450	422	18	918	738	774	562	1014
	614	594	598	802	966	114	694	578	806	146	278	866	134	690	374	642	998	722	982	930	326	242
	706	166	274	662	994	518	818	758	770	358	850	342	34	710	370	438	834	550	402	22	98	902
	118	898	742	978	726	162	70	498	822	962	934	530	406	226 470	262	50	502		102	82	86	290
	626	182	66	294	658	790	354	646	178	886	130	486	210	470	418	838	754	566	194	678	786	150
	6	306	246	258	870	338 422	854	546	198	882	950	322	38	914	534	610	390 802	434 966	630	386	230	466
	674 278	582 866	1010 134	310 690	450 374	642	18 998	918 722	738 982	774 930	562 326	1014 242	514 54	614 706	594 166	598 274	662	994	114 518	694 818	578 758	806 770
	850	342	34	710	374	438	834	550	402	22	98	902	946	118	898	742	978	726	162	70	498	822
	934	530	406	226	262	50	502	2	102	22 82	86	290	454	626	182	66	294	658	790	354	646	178
	130	486	210	470	418	838	754	566	194	678	786	150	482	6	306	246	258	870	790 338	854	546	198
	950	322	38	914	534	610	390	434	630	386	230	466	214	674	582	1010	310	450	422	18	918	738
	562	1014	514	614	594	598	802	966	114	694	578	806	146	278	866	134	690	374	642	998	722	982
	326	242	54	706	166	274	662	994	518	818	758	770	358	850	342	34	710	370	438	834	550	402
	98	902	946	118	898	742	978	726	162	70	498	822	962	934	530	406	226	262	50	502	2	102
	86	290 150	454 482	626	182 306	66 246	294 258	658 870	790 338	354 854	646 546	178 198	886 882	130 950	486 322	210 38	470 914	418 534	838 610	754 390	566 434	194 630
	786 230	466	214	6 674	582	1010	310	450	422	18	918	738	774	562	1014	514	614	594 594	598	802	966	114
	578	806	146	278	866	134	690	374	642	998	722	982	930	326	242	54	706	166	274	662	994	518
	758	770	358	850	342	34	710	370	438	834	550	402	22	98	902	946	118	898	742	978	726	162
	498	822	962	934	530	406	226	262	50	502		102	22 82	86	290	454	626	182	66	294	658	790
	646	178	886	130	486	210	470	418	838	754	566	194	678	786	150	482		306	246	258	870	338
	546	198	882 774	950	322	38	914	534	610	390	434	630	386	230 578	466	214	674	582	1010	310	450	422
	918	738	774	562	1014	514	614	594	598	802	966	114	694	578	806	146	278	866	134	690	374	642
	722 550	982 402	930 22	326 98	242 902	54 946	706 118	166 898	274 742	662 978	994 726	518 162	818 70	758 498	770 822	358 962	850 934	342 530	34 406	710 226	370 262	438 50
	2 2	102	82 82	96 86	290	454	626	182	66	294	658	790	354	646	178	902 886	130	486	210	470	418	838
	566	194	678	786	150	482	6	306	246	258	870	338	854	546	198	882	950	322	38	914	534	610
	434	630	386	230	466	214	674	582	1010	310	450	422	18	918	738	774	562	1014	514	614	594	598
	966	114	694	578	806	146	278	866	134	690	374	642	998	722 550	982	930	326	242	54	706	166	274
	994	518	818	758	770	358	850	342	34	710	370	438	834		402	22	98	902	946	118	898	742
	726	162	70	498	822	962	934	530	406	226	262	50	502	2	102	82	86	290	454	626	182	66
	658 870	790	354	646	178	886 882	130 950	486	210	470	418	838	754	566	194	678	786	150	482	6	306	246
	870	338	854	546	198	882	950															

随后, 我发现, 确实存在周期。

从 n=20 开始, f(n)为一个周期为 128 的周期数列.

 $930 \quad 326 \quad 242 \quad 54 \quad 706 \quad 166 \quad 274 \quad 662 \quad 994 \quad 518 \quad 818 \quad 758 \quad 770 \quad 358 \quad 850 \quad 342 \quad 34 \quad 710$ $370 \quad 438 \quad 834 \quad 550 \quad 402 \quad 22 \quad 98 \quad 902 \quad 946 \quad 118 \quad 898 \quad 742 \quad 978 \quad 726 \quad 162 \quad 70 \quad 498 \quad 749 \quad 7$ 822 $962 \quad 934 \quad 530 \quad 406 \quad 226 \quad 262 \quad 50 \quad 502 \quad 2 \quad 102 \quad 82 \quad 86 \quad 290 \quad 454 \quad 626 \quad 182 \quad 66$ 294 658 790 354 646 178 886 130 486 210 470 418 838 754 566 194 678 786 150 $482 \quad 6 \quad 306 \quad 246 \quad 258 \quad 870 \quad 338 \quad 854 \quad 546 \quad 198 \quad 882 \quad 950 \quad 322$ 38 914 534 610 390 $434 \quad 630 \quad 386 \quad 230 \quad 466 \quad 214 \quad 674 \quad 582 \quad 1010 \quad 310 \quad 450 \quad 422 \quad 18$ 918 738 774562 1014 514 614 594 598 802 966 114 694 578 806 146 278 866 134 690 374 642 998 722 982

因此,我们可以利用周期性。对于 n>=20 的数,先对其进行减 20, mode 128, 再加 20 的操作。对于 n<20 的数,先不做操作。

然后,可以利用优化方案①中的程序求解。

程序如下:

```
.ORIG x3000
          ADD R1, R1, #1
          ADD R2, R2, #1
ADD R3, R3, #2
LD R5, VALUE
          ADD R6, R0, R5
          BRn NEXT
          ADD R0, R0, R5
          LD R6, MOD1
          AND RO, RO, R6
LD R6, VALUE2
          ADD RO, RO, R6
NEXT
          LD R5, MOD
          ADD R7, R7, R0
ADD R0, R0, #-2
          BRnz STOP
          ADD R7, R3, R1
ADD R7, R7, R1
AGAIN2
          ADD R1, R2, #0
ADD R2, R3, #0
          ADD R3, R7, #0
ADD R0, R0, #-1
          BRp AGAIN2
          AND R7, R7, R5
STOP
          HALT
VALUE
           .FILL #-20
          .FILL #20
VALUE2
MOD1
           .FILL #127
MOD
          .FILL #1023
          .FILL #930
Fb
          .FILL #1
Fc
          .FILL #1
Fd
          .FILL #306
.END
```

(指令执行条数分析方法同上)

测试数据:

f(24)=706 由于 24>20,需要进行减 20, mode 128,再加 20 的操作.之后与优化方案①

中的执行流程一样。因此执行指令条数为:162+8=170

指令数: 170

同理得: 指令数: 1010 f(144)=642f(456) = 66同理得: 指令数: 506 f(1088)=2同理得: 指令数: 450 f(1092)=290同理得: 指令数: 478 f(2096)=898 同理得: 指令数: 338 f(4200)=322同理得: 指令数: 730 f (8192)=514 同理得: 指令数: 898 f(12000)=258 同理得:指令数:674 f(14000)=898 同理得:指令数:338

平均指令条数为:559.2

优化比率为: 559.2/38950.8≈1.4%

总结:通过最终方案,得到优化比率为559.2/38950.8≈1.4%