Phase_6

2020年4月28日 22:08

通过划线厘清结构似乎很是方便

```
4010f4: 41 56                   push   %r14
4010f6: 41 55                   push   %r13
4010f8: 41 54                   push   %r12
4010fa: 55                      push   %rbp
4010fb: 53                      push   %rbx
4010fc: 48 83 ec 50             sub    $0x50,%rsp
401100: 49 89 e5                mov    %rsp,%r13
401103: 48 89 e6                mov    %rsp,%rsi
401106: e8 51 03 00 00          callq  40145c <read_six_numbers>
40110b: 49 89 e6                mov    %rsp,%r14
40110e: 41 bc 00 00 00 00       mov    $0x0,%r12d
401114: 4c 89 ed                mov    %r13,%rbp
401117: 41 8b 45 00             mov    0x0(%r13),%eax
40111b: 83 e8 01                sub    $0x1,%eax
40111e: 83 f8 05                cmp    $0x5,%eax
401121: 76 05                   jbe    401128 <phase_6+0x34>
401123: e8 12 03 00 00          callq  40143a <explode_bomb>
401128: 41 83 c4 01             add    $0x1,%r12d
40112c: 41 83 fc 06             cmp    $0x6,%r12d
401130: 74 21                   je     401153 <phase_6+0x5f>
401132: 44 89 e3                mov    %r12d,%ebx
401135: 48 63 c3                movslq %ebx,%rax
401138: 8b 04 84                mov    (%rsp,%rax,4),%eax
40113b: 39 45 00                cmp    %eax,0x0(%rbp)
40113e: 75 05                   jne    401145 <phase_6+0x51>
401140: e8 f5 02 00 00          callq  40143a <explode_bomb>
401145: 83 c3 01                add    $0x1,%ebx
401148: 83 fb 05                cmp    $0x5,%ebx
40114b: 7e e8                   jle    401135 <phase_6+0x41>
40114d: 49 83 c5 04             add    $0x4,%r13
401151: eb c1                   jmp    401114 <phase_6+0x20>
401153: 48 8d 74 24 18          lea    0x18(%rsp),%rsi #循环结束标记
401158: 4c 89 f0                mov    %r14,%rax
40115b: b9 07 00 00 00          mov    $0x7,%ecx
401160: 89 ca                   mov    %ecx,%edx
401162: 2b 10                   sub    (%rax),%edx
401164: 89 10                   mov    %edx,(%rax)
401166: 48 83 c0 04             add    $0x4,%rax
40116a: 48 39 f0                cmp    %rsi,%rax
40116d: 75 f1                   jne    401160 <phase_6+0x6c>
40116f: be 00 00 00 00          mov    $0x0,%esi
401174: eb 21                   jmp    401197 <phase_6+0xa3>
401176: 48 8b 52 08             mov    0x8(%rdx),%rdx
40117a: 83 c0 01                add    $0x1,%eax
40117d: 39 c8                   cmp    %ecx,%eax
40117f: 75 f5                   jne    401176 <phase_6+0x82>
401181: eb 05                   jmp    401188 <phase_6+0x94>
401183: ba d0 32 60 00          mov    $0x6032d0,%edx
401188: 48 89 54 74 20          mov    %rdx,0x20(%rsp,%rsi,2)
40118d: 48 83 c6 04             add    $0x4,%rsi
401191: 48 83 fe 18             cmp    $0x18,%rsi
401195: 74 14                   je     4011ab <phase_6+0xb7>
401197: 8b 0c 34                mov    (%rsp,%rsi,1),%ecx
40119a: 83 f9 01                cmp    $0x1,%ecx
40119d: 7e e4                   jle    401183 <phase_6+0x8f>
40119f: b8 01 00 00 00          mov    $0x1,%eax
4011a4: ba d0 32 60 00          mov    $0x6032d0,%edx
4011a9: eb cb                   jmp    401176 <phase_6+0x82>
4011ab: 48 8b 5c 24 20          mov    0x20(%rsp),%rbx
4011b0: 48 8d 44 24 28          lea    0x28(%rsp),%rax
4011b5: 48 8d 74 24 50          lea    0x50(%rsp),%rsi
4011ba: 48 89 d9                mov    %rbx,%rcx
4011bd: 48 8b 10                mov    (%rax),%rdx
4011c0: 48 89 51 08             mov    %rdx,0x8(%rcx)
4011c4: 48 83 c0 08             add    $0x8,%rax
4011c8: 48 39 f0                cmp    %rsi,%rax
4011cb: 74 05                   je     4011d2 <phase_6+0xde>
4011cd: 48 89 d1                mov    %rdx,%rcx
4011d0: eb eb                   jmp    4011bd <phase_6+0xc9>
```

六个数全部变为7-自身

	Num6
	Num5
	Num4
	Num3
	Num2
%rsp,%r13,%r14	Num1

%r12d = 0

根据画的线我们可以轻易得出这是一个二重嵌套循环，中间夹杂着分支与break:

```
for(r14=r13=rsp;r12d=0;true;r13+=4){
    rbp=r13;
    if(6<(r13))
        explode;
    r12d++;
    if(r12d==6)
        break;
    for(ebx=r12d;ebx<=5;ebx++)
        if((rbp)==rsp+4*ebx))
            explode;
}
```

大意就是六个数均在1-6之间且互不相等

黑色箭头的小循环使用极其清奇的方法（链表与数组的奇异结合）：

链表按数组的方式存放

有颜色的值就是每次的rdx与8（rdx）,循环次数ecx也就是六个数的值

作用是找到链表的第ecx个节点。下表是内存情况

地址	节点值	编号	next指针
0x6032d0	0x0000014c	0x00000001	0x00000000006032e0
0x6032e0	0x000000a8	0x00000002	0x00000000006032f0
0x6032f0	0x0000039c	0x00000003	0x0000000000603300
0x603300	0x000002b3	0x00000004	0x0000000000603310
0x603310	0x000001dd	0x00000005	0x0000000000603320
0x603320	0x000001bb	0x00000006	0x0000000000000000

先执行黄色的循环（jump to middle)六个数

如果为1，就向栈里填入0x6032d0（头节点）

否则进入上方黑色直线箭头小循环，将得到的结果入栈

全部都小于等于1，就直接到判断部分

将上图的指针按输入顺序重排

也即将链表按输入的顺序重排

如我输入了5 4 3 6 1，变换后为2 3 4 1 6

链表就按编号23416重排

检测链表是否是递减

```
4011c8: 48 39 f0                cmp    %rsi,%rax
4011cb: 74 05                   je     4011d2 <phase_6+0xde>
4011cd: 48 89 d1                mov    %rdx,%rcx
4011d0: eb eb                   jmp    4011bd <phase_6+0xc9>
4011d2: 48 c7 42 08 00 00 00    movq   $0x0,0x8(%rdx)
4011d9: 00
4011da: bd 05 00 00 00          mov    $0x5,%ebp
4011df: 48 8b 43 08             mov    0x8(%rbx),%rax
4011e3: 8b 00                   mov    (%rax),%eax
4011e5: 39 03                   cmp    %eax,(%rbx)
4011e7: 7d 05                   jge    4011ee <phase_6+0xfa>
4011e9: e8 4c 02 00 00          callq  40143a <explode_bomb>
4011ee: 48 8b 5b 08             mov    0x8(%rbx),%rbx
4011f2: 83 ed 01                sub    $0x1,%ebp
4011f5: 75 e8                   jne    4011df <phase_6+0xeb>
4011f7: 48 83 c4 50             add    $0x50,%rsp
4011fb: 5b                      pop    %rbx
4011fc: 5d                      pop    %rbp
4011fd: 41 5c                   pop    %r12
4011ff: 41 5d                   pop    %r13
401201: 41 5e                   pop    %r14
401203: c3                      retq
```

检测链表是否是递减