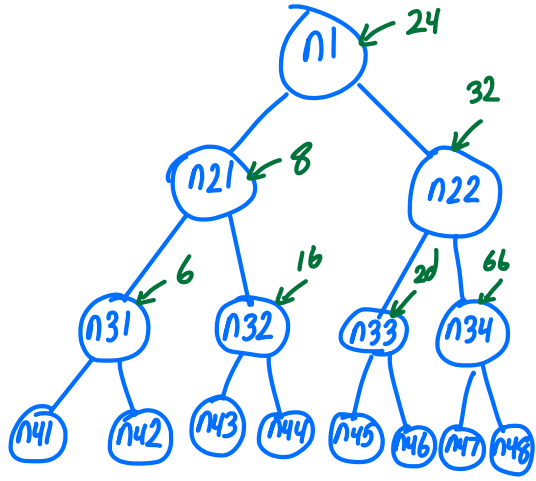
```
Breakpoint 1, 0x000055555555a1b in secret_phase ()
(gdb) disas
Dump of assembler code for function secret_phase:
=> 0x000055555555a1b <+0>:     endbr64
   0x000055555555a1f <+4>:     push   %rbx
   0x000055555555a20 <+5>:     call   0x555555555e96 <read_line>
   0x000055555555a25 <+10>:    mov    %rax,%rdi
   0x000055555555a28 <+13>:    mov    $0xa,%edx
   0x000055555555a2d <+18>:    mov    $0x0,%esi
   0x000055555555a32 <+23>:    call   0x555555555310 <strtol@plt>
   0x000055555555a37 <+28>:    mov    %eax,%ebx
   0x000055555555a39 <+30>:    sub    $0x1,%eax
   0x000055555555a3c <+33>:    cmp    $0x3e8,%eax
   0x000055555555a41 <+38>:    ja     0x555555555a69 <secret_phase+78>
   0x000055555555a43 <+40>:    mov    %ebx,%esi
   0x000055555555a45 <+42>:    lea    0x3b04(%rip),%rdi        # 0x555555559550
   0x000055555555a4c <+49>:    call   0x5555555559da <fun7>
   0x000055555555a51 <+54>:    cmp    $0x6,%eax
   0x000055555555a54 <+57>:    jne    0x555555555a70 <secret_phase+85>
   0x000055555555a56 <+59>:    lea    0x1723(%rip),%rdi        # 0x555555557180
   0x000055555555a5d <+66>:    call   0x555555555250 <puts@plt>
   0x000055555555a62 <+71>:    call   0x555555555fce <phase_defused>
   0x000055555555a67 <+76>:    pop    %rbx
--Type <RET> for more, q to quit, c to continue without paging--
   0x000055555555a68 <+77>:    ret
   0x000055555555a69 <+78>:    call   0x555555555e0f <explode_bomb>
   0x000055555555a6e <+83>:    jmp    0x555555555a43 <secret_phase+40>
   0x000055555555a70 <+85>:    call   0x555555555e0f <explode_bomb>
   0x000055555555a75 <+90>:    jmp    0x555555555a56 <secret_phase+59>
End of assembler dump.
(gdb)
```

we want %eax to be below or equal to

3e8 = 1000

```
(gdb) x/16dx 0x555555559550
0x555555559550 <n1>:    0x00000024      0x00000000      0x55559570      0x00005555
0x555555559560 <n1+16>: 0x55559590      0x00005555      0x00000000      0x00000000
0x555555559570 <n21>:   0x00000008      0x00000000      0x555595f0      0x00005555
0x555555559580 <n21+16>:        0x555595b0      0x00005555      0x00000000      0x00000000
(gdb)
```

```
(gdb) x/8dx 0x555555559550
[0x555555559550 <n1>:    0x00000024      0x00000000      0x55559570      0x00005555
 0x555555559560 <n1+16>: 0x55559590      0x00005555      0x00000000      0x00000000
(gdb) x/16dx 0x555555559550
 0x555555559550 <n1>:    0x00000024      0x00000000      0x55559570      0x00005555
 0x555555559560 <n1+16>: 0x55559590      0x00005555      0x00000000      0x00000000
[0x555555559570 <n21>:   0x00000008      0x00000000      0x555595f0      0x00005555
 0x555555559580 <n21+16>:        0x555595b0      0x00005555      0x00000000      0x00000000
(gdb) x/60gx 0x555555559550
 0x555555559550 <n1>:    0x0000000000000024      0x0000555555559570
 0x555555559560 <n1+16>: 0x0000555555559590      0x0000000000000000
 0x555555559570 <n21>:   0x0000000000000008      0x00005555555595f0
 0x555555559580 <n21+16>:        0x00005555555595b0      0x0000000000000000
 0x555555559590 <n22>:   0x0000000000000032      0x0000555555559d0
 0x5555555595a0 <n22+16>:        0x0000555555559610      0x0000000000000000
[0x5555555595b0 <n32>:   0x0000000000000016      0x00005555555590c0
 0x5555555595c0 <n32+16>:        0x0000555555559080      0x0000000000000000
 0x5555555595d0 <n33>:   0x000000000000002d      0x0000555555559020
 0x5555555595e0 <n33+16>:        0x00005555555590e0      0x0000000000000000
 0x5555555595f0 <n31>:   0x0000000000000006      0x0000555555559040
 0x555555559600 <n31+16>:        0x00005555555590a0      0x0000000000000000
 0x555555559610 <n34>:   0x000000000000006b      0x0000555555559060
 0x555555559620 <n34+16>:        0x0000555555559100      0x0000000000000000
 0x555555559630 <node1>: 0x0000000100000091      0x0000555555559660
 0x555555559640 <node2>: 0x000000020000012d      0x0000555555559630
 0x555555559650 <node3>: 0x00000003000001bb      0x0000555555559670
 0x555555559660 <node4>: 0x000000040000003c      0x0000000000000000
 0x555555559670 <node5>: 0x0000005000001af       0x0000555555559640
 0x555555559680 <host_table>:    0x000055555555745f      0x0000555555557470
 0x555555559690 <host_table+16>: 0x0000000000000000      0x0000000000000000
 0x5555555596a0 <host_table+32>: 0x0000000000000000      0x0000000000000000
 0x5555555596b0 <host_table+48>: 0x0000000000000000      0x0000000000000000
--Type <RET> for more, q to quit, c to continue without paging--
```

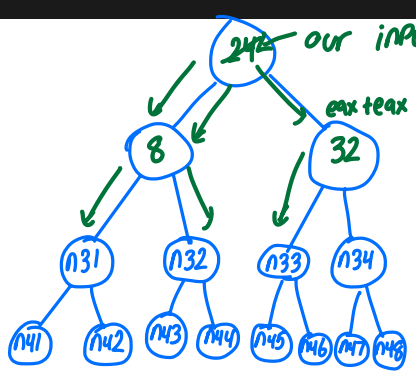



```
tinaj — ssh jcountry@pi.cs.oswego.edu — 80×24

Dump of assembler code for function fun7:
=> 0x00005555555559da <+0>:     endbr64
   0x00005555555559de <+4>:     test   %rdi,%rdi      <- checks if there is a new node to go to
   0x00005555555559e1 <+7>:     je     0x555555555a15 <fun7+59>
   0x00005555555559e3 <+9>:     sub    $0x8,%rsp
   0x00005555555559e7 <+13>:    mov    (%rdi),%edx    <- Turns %edx at value at node
   0x00005555555559e9 <+15>:    cmp    %esi,%edx      <- checks if edx (value at node) > our input
   0x00005555555559eb <+17>:    jg     0x555555559f9 <fun7+31>
   0x00005555555559ed <+19>:    mov    $0x0,%eax      If not, mov 0 to %eax
   0x00005555555559f2 <+24>:    jne    0x555555555a06 <fun7+44>
   0x00005555555559f4 <+26>:    add    $0x8,%rsp
   0x00005555555559f8 <+30>:    ret
   0x00005555555559f9 <+31>:    mov    0x8(%rdi),%rdi
   0x00005555555559fd <+35>:    call   0x55555555559da <fun7>
   0x0000555555555a02 <+40>:    add    %eax,%eax
   0x0000555555555a04 <+42>:    jmp    0x55555555559f4 <fun7+26>
   0x0000555555555a06 <+44>:    mov    0x10(%rdi),%rdi
   0x0000555555555a0a <+48>:    call   0x55555555559da <fun7>
   0x0000555555555a0f <+53>:    lea    0x1(%rax,%rax,1),%eax
   0x0000555555555a13 <+57>:    jmp    0x55555555559f4 <fun7+26>
   0x0000555555555a15 <+59>:    mov    $0xffffffff,%eax  <- return -1 (Node not found)
   0x0000555555555a1a <+64>:    ret                      ☆ More than 1 solution
End of assembler dump.
(gdb)
```

so input of 24 would make %eax=0

%edi → 0x555555559550
%esi → our input as a long
%edx → value at node



32-8 = 24