CSC407

Haonan(Harry Chen)

Homework1

1. C programming

See harry.zip for detail.

1. Timing Part1(with no optimization)

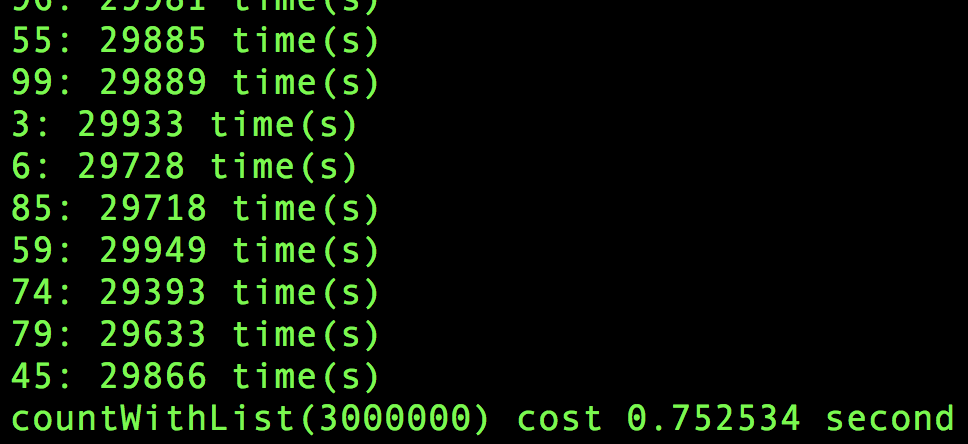
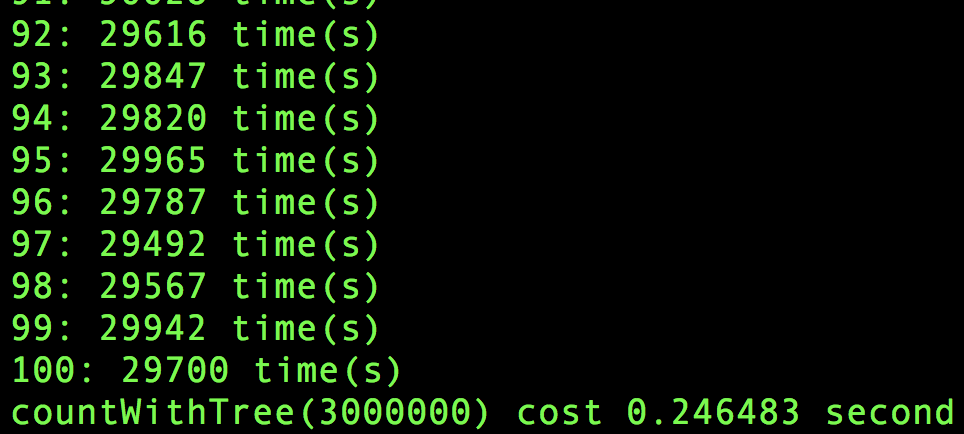
Here is the report of Function Call without optimization



GenerateList cost 0.06 self second and GenerateTree cost 0.02 self second without optimization.

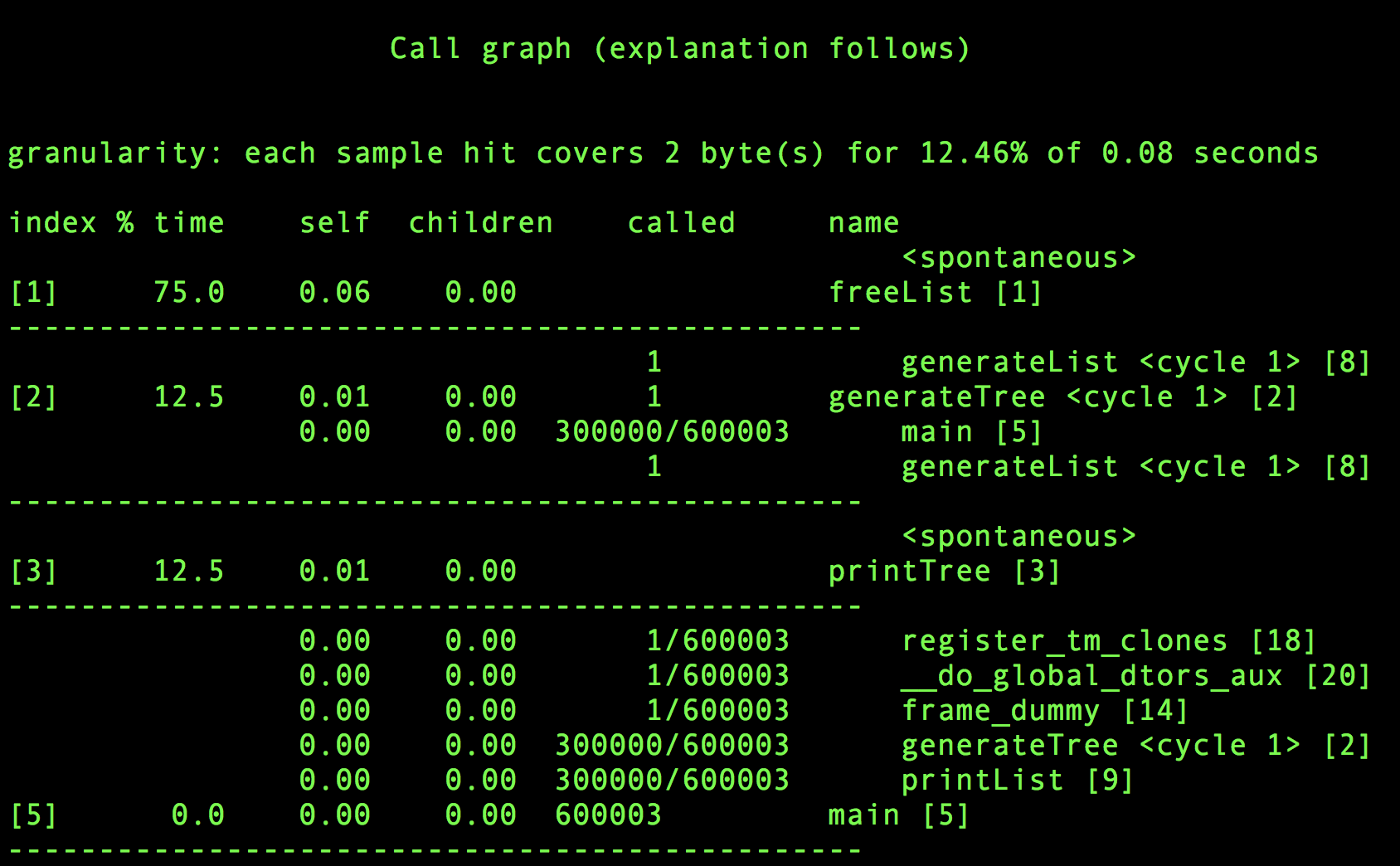
I also write a “timer function” embed in the main function, here are the results

countWithList(3,000,000) countWithTree(3,000,000)

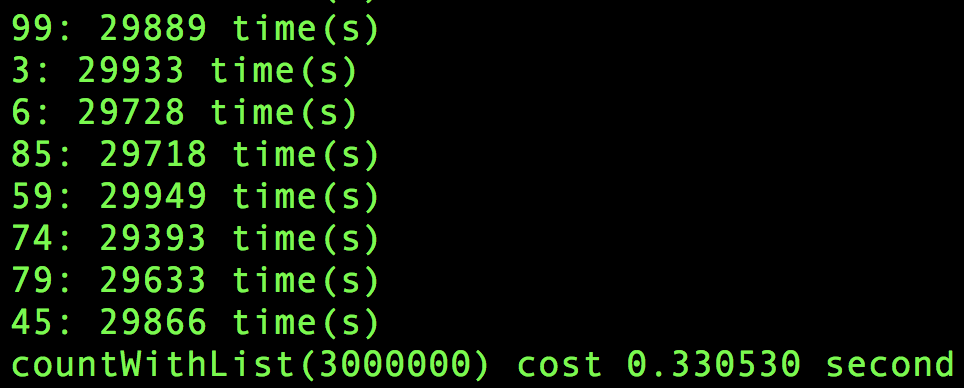
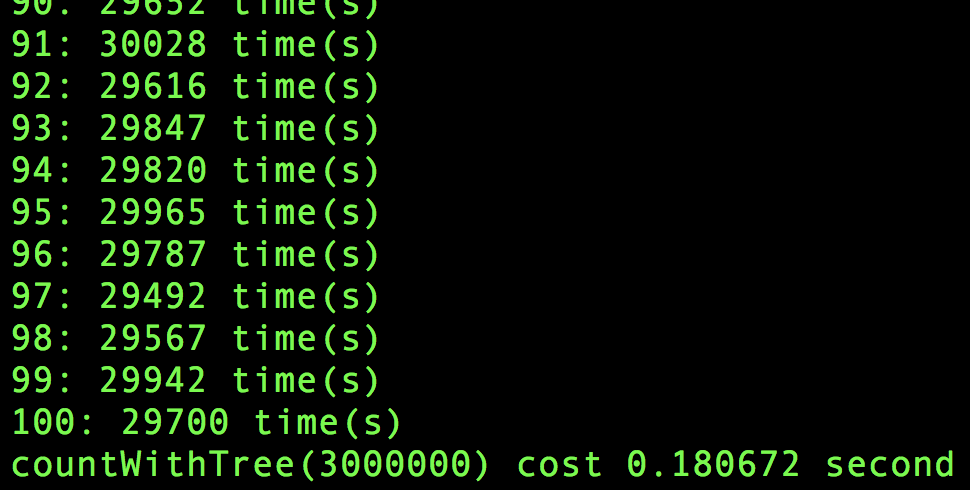
 

1. Timing Part2(with optimization)

Here is the report of Function Call with optimization



countWithList(3,000,000) countWithTree(3,000,000)

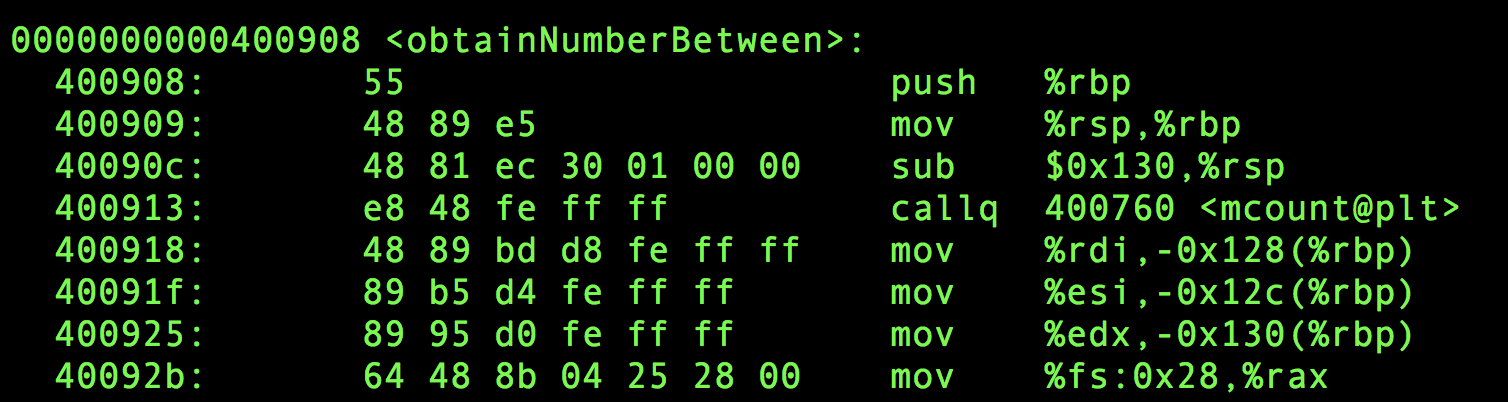
 

1. Parts of an executable

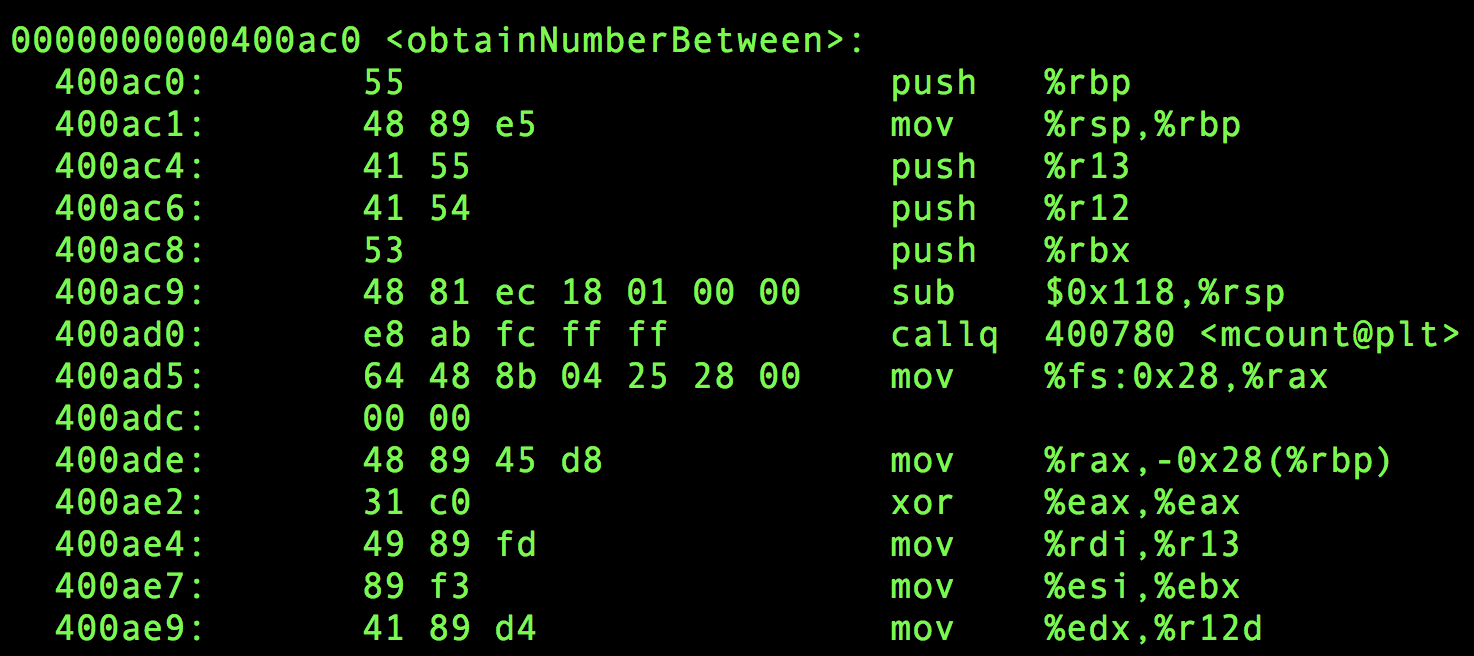
|  |  |  |
| --- | --- | --- |
| **Question** | **Command** | **Result** |
| **(A)** | **CANNOT BE FOUND** | **Entry is a local variable in `obtainNumberBetween()` function, it**  **will be save in %eax register in runtime.** |
| **(B)** | **objdump -s -j .rodata assign1-0 | grep 'What'** |  |
| **(C)** | **objdump -d -j .text assign1-0** |  |
| **(D)** | **objdump -t -j .bss assign1-0** |  |

1. Compare optimizations.
2. Save in register rather than in memory

Non-optimization version use memory to save the value of 3 arguments includes descriptionCPtr, low and high in `obtainNumberBetween()` function.

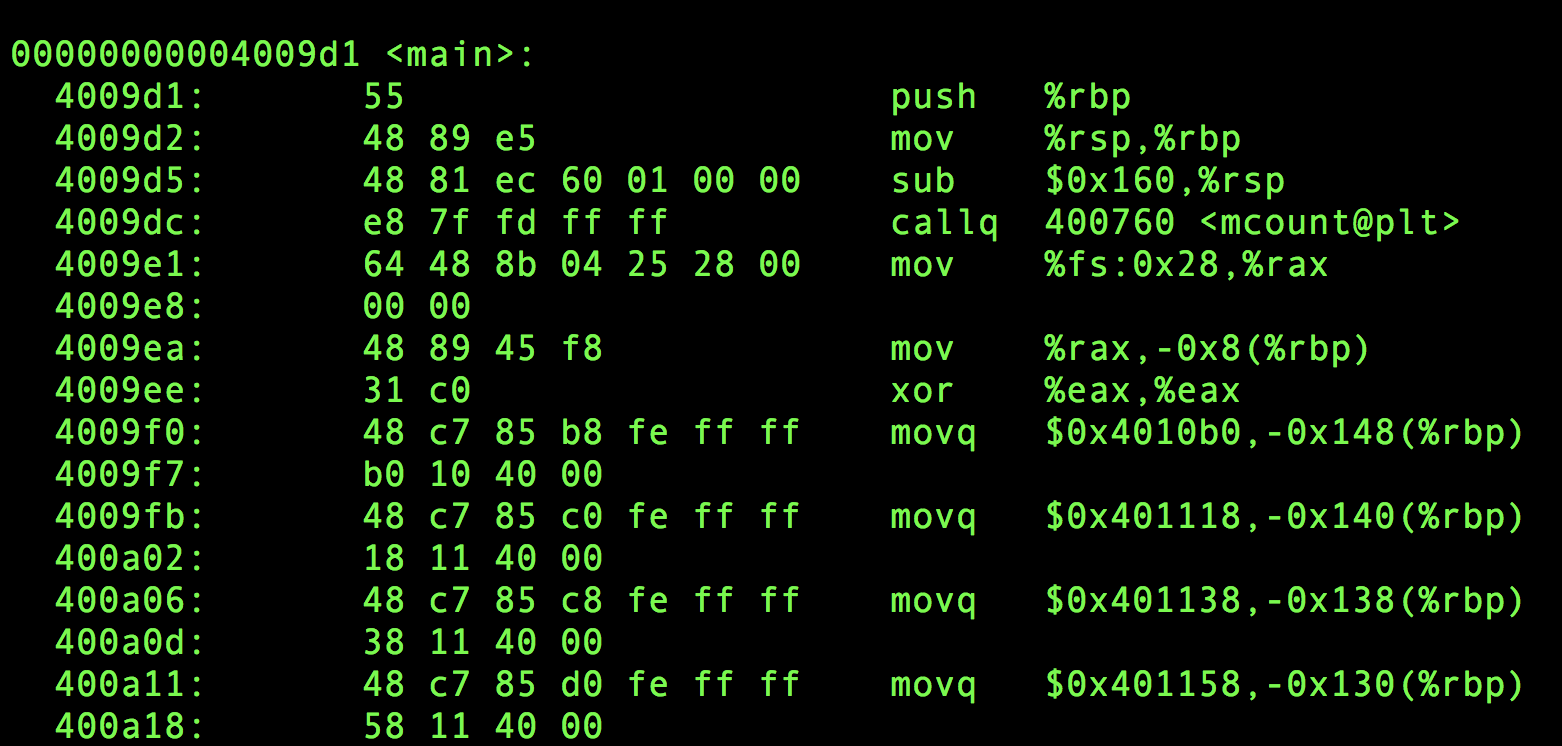


Optimization version use register to save the value.



1. There is another optimization example of using register rather than memory

Assign1-0 use memory



Assign1-2 use register

