0816028 周孟謙

Instructions executed

Fibonacci: 125

Gcd: 75

Bubble sort:

main 的 la a1,str1 到 jal ra,printstring

print 的 li t0,0 到 lw t1,N

print_for 從頭到尾跑 10 次後從 bge t0,t1,print_end 跳到 print_end

bge t0,t1,printArray_for_End 從頭到尾跑一次

跑 bubblesort

跑 bubble_out 跑十次:直到 t0>t2 時跑 out_end

在 bubble_out 裡面，跑 bubble_in 跑直到 t1<0 或 t5>=t4 時跑 in_end

然後回 bubble_out，若無 t1<0 或 t5>=t4，則跑 swap 再跳回

bubble_ine 的頭

跑 printfstring 從頭到尾

從 la a1,str2 到 jal ra,printstring

print 的 li t0,0 到 lw t1,N

print_for 從頭到尾跑 10 次後從 bge t0,t1,print_end 跳到 print_end

結束程式

Max number push into stack

Fibonacci: 21

Gcd: 2

Bubble sort: 4

心得:

這次的作業有點難，花了整整兩天假日把指令和運作原理弄懂，一開始真的有點手足無措，不過當寫出來第一個費撥那作業時真的讓人信心滿滿，費撥那我是以原本給的範例 fact 做修改後寫完，把原本每次放入 stack 的數量改成三個: 分別為 ra, 下一個費撥那數和目前的費撥那數，這樣下次只要把上次的下一個費撥那數改成現在的，然後再把上次的下一個費撥那數和目前的費撥那數加起來，就變成目前的下一個費撥那數。Gcd 則是直接把 c++的 code 的邏輯直接搬到 ripe 上，比較需要注意的是程式裡面 ret 的運作方式是以 ra 記住的位置來跑，其他沒遇到什麼困難。後來也以此氣勢寫出最後的 bubble sort，這次的作業讓我把 RISC-V 的指令運用從零到有學會，也獲得了莫大的成

就感。