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。