

Báo Cáo Thực Hành

Môn: Thực Hành Kiến Trúc Máy Tính

**Assignment 1:**

**Code:**

*.data*

*A: .word -2, 6, -1, 3, -2*

*.text*

*main:*

*la $a0, A*

*li $a1, 5*

*j mspfx*

*nop*

*continue:*

*lock:*

*j lock # stop program*

*nop*

*end\_of\_main:*

*# proceduce mspfx*

*mspfx:*

*addi $v0, $zero, 0 # initialize length in $v0 to 0*

*addi $v1, $zero, 0 # initialize max sum in $v1to 0*

*addi $t0, $zero, 0 # initialize index i in $t0 to 0*

*addi $t1, $zero, 0 # initialize running sum in $t1 to 0*

*loop:*

*add $t2, $t0, $t0 # put 2i in $t2*

*add $t2, $t2, $t2 # put 4i in $t2*

*add $t3, $t2, $a0 # put 4i + A(address of A[i])*

*lw $t4, 0($t3) # load A[i] into $t4*

*add $t1, $t1, $t4 # $t1 = sum += A[i]*

*slt $t5, $v1, $t1*

*bne $t5, $zero, mdfy # if max sum < new sun*

*j test*

*mdfy:*

*addi $v0, $t0, 1*

*addi $v1, $t1, 0*

*test:*

*addi $t0, $t0, 1*

*slt $t5, $t0, $a1*

*bne $t5, $zero, loop*

*done: j continue*

*mspfx\_end:*

**Kết quả:**



Mảng A = {-2, 6, -1, 3, -2}

Length = $v0 = 4

Khi đó: max\_sum = $v1 =. 2 + 6 + (-1) + 3 = 6

**Assignment 2:**

**Code:**

*.data*

*A: .word 7, -2, 5, 1, 5,6,7,3,6,8,8,59,5*

*Aend: .word*

*.text*

*main:*

*la $a0,A # $a0 = Address(A[0])*

*la $a1,Aend*

*addi $a1,$a1,-4 # $a1 = Address(A[n-1])*

*j sort # sort*

*after\_sort:*

*li $v0, 10 #exit*

*syscall*

*end\_main:*

*#--------------------------------------------------------------*

*#procedure sort (ascending selection sort using pointer)*

*#register usage in sort program*

*#$a0 pointer to the first element in unsorted part*

*#$a1 pointer to the last element in unsorted part*

*#$t0 temporary place for value of last element*

*#$v0 pointer to max element in unsorted part*

*#$v1 value of max element in unsorted part*

*#--------------------------------------------------------------*

*sort:*

*beq $a0,$a1,done #single element list is sorted*

*j max #call the max procedure*

*after\_max:*

*lw $t0,0($a1) #load last element into $t0*

*sw $t0,0($v0) #copy last element to max location*

*sw $v1,0($a1) #copy max value to last element*

*addi $a1,$a1,-4 #decrement pointer to last element*

*j sort #repeat sort for smaller list*

*done: j after\_sort*

*#------------------------------------------------------------------------*

*#Procedure max*

*#function: fax the value and address of max element in the list*

*#$a0 pointer to first element*

*#$a1 pointer to last element*

*#------------------------------------------------------------------------*

*max:*

*addi $v0,$a0,0 #init max pointer to first element*

*lw $v1,0($v0) #init max value to first value*

*addi $t0,$a0,0 #init next pointer to first*

*loop:*

*beq $t0,$a1,ret #if next=last, return*

*addi $t0,$t0,4 #advance to next element*

*lw $t1,0($t0) #load next element into $t1*

*slt $t2,$t1,$v1 #(next)<(max) ?*

*bne $t2,$zero,loop #if (next)<(max), repeat*

*addi $v0,$t0,0 #next element is new max element*

*addi $v1,$t1,0 #next value is new max value*

*j loop #change completed; now repeat*

*ret:*

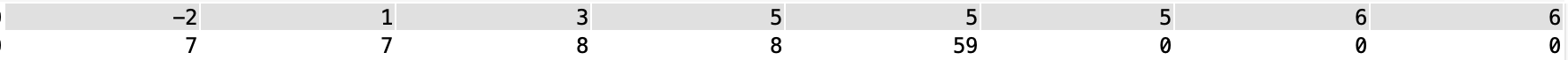
*j after\_max*

**Kết quả:**

Trước khi sắp xếp:



Sau khi sắp xếp:



**Assignment 3:**

**Algorithms:**

*loop1(i = 0; i < n; i++)*

*loop2(j = 0; j < n-i-1; j++)*

*if(A[j] > A[j+1])*

*swap(A[j], A[j+1])*

*end\_if*

*end\_loop2*

*end\_loop1*

**Code:**

*.data*

*A: .word 7, -2, 5, 1, 5,6,7,3,6,8,8,59,5*

*.text*

*main:*

*la $a0,A #$a0 = Address(A[0])*

*li $s0, 13 #length of array $s0, length*

*j sort #sort*

*after\_sort:*

*li $v0, 10 #exit*

*syscall*

*end\_main:*

*#Bubble sort algorithm*

*sort:*

*li $t0, 0 # $t0 = i = 0*

*loop1:*

*slt $v0, $t0, $s0 # set $v0 = 1 when i < length*

*beq $v0, $zero, end\_loop1 # end loop when i >= length*

*li $t1, 0 # $t1 = j = 0*

*loop2:*

*addi $t2, $s0, -1*

*sub $t2, $t2, $t0 # $t2 = temp = n-i-1*

*slt $v0, $t1, $t2 # set $v0 = 1 when j < temp*

*beq $v0, $zero, end\_loop2*

*if:*

*sll $t5, $t1, 2 # $t5 = 4\*j*

*add $t5, $t5, $a0 # $t5 is address A[j]*

*lw $t3, 0($t5) # Load $t3 = A[j]*

*lw $t4, 4($t5) # $t4 = A[j+1]*

*sgt $v0, $t3, $t4 # set $v0 = 1 when A[j] > A[j+1]*

*beq $v0, $zero, end\_if # End\_if if A[j] <= A[j+1]*

*j swap*

*end\_if:*

*addi $t1, $t1, 1 # j++*

*j loop2*

*end\_loop2:*

*addi $t0, $t0, 1 # i++*

*j loop1*

*end\_loop1:*

*j after\_sort*

*swap:*

*sw $t4, 0($t5)*

*sw $t3, 4($t5)*

*j end\_if*

**Kết quả:**

Trước khi sắp xếp:



Sau khi sắp xếp:



**Assignment 4:**

**Algorithms:**

*loop(i = 1; i < n; i++)*

*j = i – 1;*

*key = A[i];*

*while(t < A[j] && j >= 0)*

*A[j+1] = A[j];*

*j--;*

*end\_while*

*A[j+1] = key;*

*end\_loop*

**Code:**

*.data*

*A: .word 7, -2, 5, 1, 5,6,7,3,6,8,8,59,5*

*.text*

*main:*

*la $a0,A #$a0 = Address(A[0])*

*li $s0, 13 #length of array $s0, length*

*j sort #sort*

*after\_sort:*

*li $v0, 10 #exit*

*syscall*

*end\_main:*

*#Insertion sort algorithm*

*sort:*

*li $t0, 1 # $t0, i = 1*

*li $t1, 0 # $t1, key = 0*

*li $t2, 0 # $t2, j = 0*

*loop:*

*slt $v0, $t0, $s0 # set $v0 = 1 when i < length*

*beq $v0, $zero, end\_loop # end loop when i >= length*

*add $t3, $t0, $t0 # $t3 = 2\*i*

*add $t3, $t3, $t3 # $t3 = 4\*i*

*add $t3, $t3, $a0 # $t3 is address A[i]*

*lw $t1, 0($t3) # key = A[i]*

*add $t2, $t0, -1 # j = i - 1*

*while:*

*slt $v0, $t2, $zero # set $v0 = 1 when j < 0*

*bne $v0, $zero, end\_while*

*add $t3, $t2, $t2 # $t3 = 2\*j*

*add $t3, $t3, $t3 # $t3 = 4\*j*

*add $t3, $t3, $a0 # $t3 is address A[j]*

*lw $t4, 0($t3) # Load A[j]*

*slt $v0, $t1, $t4 # set $v0 = 1 when A[j] > key*

*beq $v0, $zero, end\_while # End while if key >= A[j]*

*sw $t4, 4($t3) # A[j+1] = A[j]*

*add $t2, $t2, -1 # j = j - 1*

*j while*

*end\_while:*

*add $t3, $t2, $t2 # $t3 = 2\*j*

*add $t3, $t3, $t3 # $t3 = 4\*j*

*add $t3, $t3, $a0 # $t3 is address A[j]*

*sw $t1, 4($t3) # A[j+1] = key*

*add $t0, $t0, 1 # i = i + 1*

*j loop*

*end\_loop:*

**Kết quả:**

Trước khi sắp xếp:



Sau khi sắp xếp:



**Conclusions:**

* What the advantage and disadvantage of two methods: indexing and updating pointer:
  + Advantage:
    - Dùng chỉ mục trực quan dễ hiểu.
    - Dùng con trỏ tiết kiệm tính toán.
  + Disadvantage:
    - Dùng chỉ mục cần tốn thời gian tính toán địa chỉ.
    - Dùng con trỏ khó khiểm soát và không trực quan dễ hiểu.