/\*

void qsort(int a[], int lo, int hi)

{

int i = lo, j = hi, h;

int x = a[(lo + hi) / 2];

do

{

while (a[i] < x) i++;

while (a[j] > x) j--;

if (i <= j)

{

h = a[i];

a[i] = a[j];

a[j] = h;

i++;

j--;

}

}while (i <= j)

if (lo < i) qsort(a, lo, j);

if (i < hi) qsort(a, i, hi);

}

\*/

#$a0 🡪 array, $a1 🡪 lo, $a2 🡪 hi

qsort:

addi $sp, $sp, -12 # adjust stack for 3 more items

sw $a0, 0($sp) # save $a0(array)

sw $a1, 4($sp) # save lo ($a1)

sw $a2, 8($sp) # save hi ($a2)

sw $ra, 12($sp) # save return address

move $s1, $a1 # i ($s1) = lo ($a1)

move $s2, $a2 # j ($s2) = hi ($a2)

add $t1, $a1, $a2 # $t1 = lo ($a1) + hi ($a2)

sll $t1, $t1, 1 # $t1 = [ (hi+lo) / 2] \* 4 bytes

add $t1, $a0, $t1 # $t1 = $a0 + $t1

lw $t2, 0($t1) # x($t2) = a[(lo + hi) / 2]

loop1: # while(a[i] < x) i++;

sll $t3, $s1, 2 # $t3 = $s1 \* 4 bytes

add $t3, $a0, $t3 # $t3 = $a0 + $t3

lw $t3, 0($t3) # $t3 = a[i]

slt $t5, $t3, $t2 # if a[i] ($t3) < x ($t2), $t5 = 1; else 0

beq $t5, $zero, loop2 # if a[i] >= x ($t5 == 0), end loop1

addi $s1, $s1, 1 # i ($s1)++

j loop1 # run loop1 again

loop2: # while (a[j] > x) j--;

sll $t4, $s2, 2 # $t4 = $s2 \* 4 bytes

add $t4, $a1, $t4 # $t4 = $a1 + $t4

lw $t4, 0($t4) # $t4 = a[j]

slt $t5, $t2, $t4 # if x ($t2) < a[j] ($t4), $t5 = 1; else $t5 = 0

beq $t5, $zero, endloop2 # if a[j] <= x ($t5 == 0), end loop2

addi $s2, $s2, -1 # j = j - 1

j loop2 # run loop2 again

endloop2:

slt $t5, $s2, $s1 # if i ($s1) <= j ($s2), $t5 = 0; else $t5 = 1

bne $t5, $zero, dowhile # if i <= j is false ($t5 = 1), jump to dowhile

move $t5, $t3 #h($t5) = a[i] ($t3)

move $t3, $t4 #a[i] ($t3) = a[j] ($t4)

move $t4, $t5 #a[j] ($t4) = h ($t5)

addi $s1, $s1, 1 # i ($s1) ++

addi $s2, $s2, -1 # j ($s2) - -

dowhile:

slt $t5, $s2, $s1 # if i <= j is false, $t5 = 1; True $t5 = 0

beq $t5, $zero, loop1 # if i <= j is True ($t5 == 0), run loop1 again

if1:

slt $t5, $a1, $s2 # if lo ($a1) < j ($a2), $t5 = 1; else $t5 = 0

beq $t5, $zero, if2 # if lo < j is False (t5 == 0), jump to if2

lw $a1, 4($sp) # $a1 = lo (4($sp))

move $a2, $s2 # $a2 = j ($s2)

jal qsort # recall qsort

if2:

slt $t5, $s1, $a2 # if i < hi is True, $t5 = 1; False $t5 = 0

beq $t5, $zero, exit # if I < hi is False ($t5 == 0), exit

lw $a2, 8($sp) # $a2 = hi ($a1)

move $a1, $s1 # $a1 = i ($s1)

jal qsort # recall qsort

exit:

lw $a0, 0($sp) # restore $a0

lw $a1, 4($sp) # restore $a1

lw $a2, 8($sp) # restore $a2

lw $ra, 12($sp) # restore return address

addi $sp, $sp, 12 # restore the stack

jr $ra #return