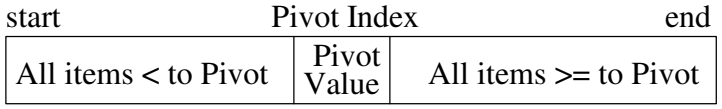
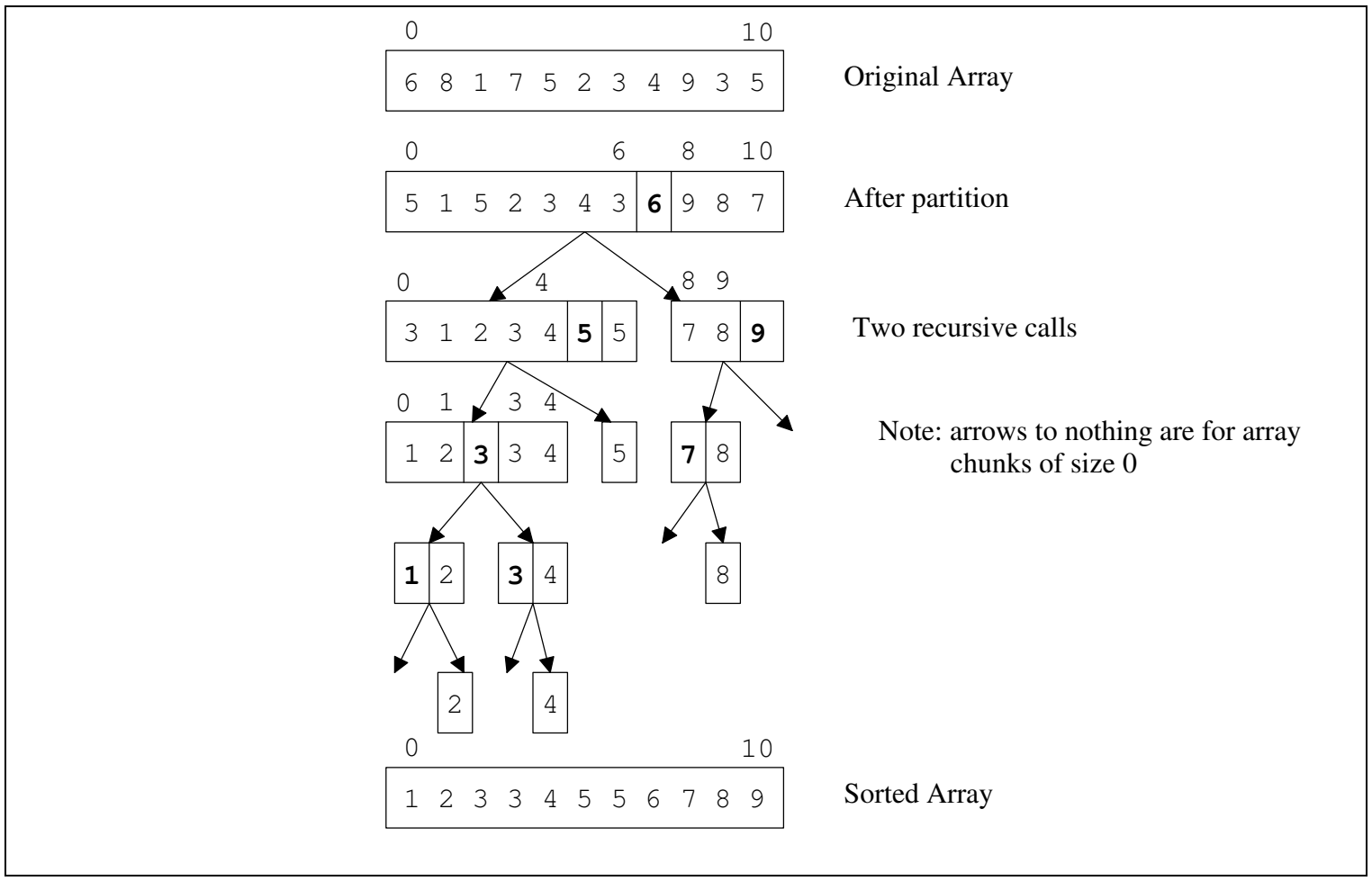
Homework #7 Computer Organization
Due: April 20, 2019 (Sat.) by 11:59 PM

Quick sort is a sorting algorithm that uses recursion (i.e., calls itself). The general idea is as follows.

1. Rearrange (called *partitioning*) the unsorted items by:
   a) Selecting a "random" item as the *pivot*, and
   b) Rearranging items as shown in the diagram
2. Quick sort the unsorted part to the left of the pivot
3. Quick sort the unsorted part to the right of the pivot



Quick sort repeatedly tackles smaller chunks of the array until a base case -- when array chunk of size 1 or 0.



Translate the following quick sort algorithm to MIPS assembly language. **(YOU ARE TO USE THE MIPS CALLING CONVENTIONS WHEN IMPLEMENTING THE QUICK SORT, PARTITION, AND SWAP SUBPROGRAMS!!!)**

Use the data below when you run your program.

```
        .data
array:  .word 6 8 1 7 5 2 3 4 9 3 5
length: .word 11

        .text
        .globl  main
main:
      # call quickSort here using $a register parameters

      li    $v0, 10           # system code for exit
      syscall
```

```
main( )
begin
    quickSort(array, 0, length-1)
end main


quickSort( start address of integer array,
               integer  start, end)
local variable
    integer pivotIndex
begin
    if (start < end) then
        pivotIndex = partition(array, start, end)
        quickSort(array, start, pivotIndex-1)
        quickSort(array, pivotIndex+1, end)
    end if
end quickSort
```

(swap should exchange the memory values pointed at by the two addresses passed it)

```
swap( address of integer operand1,
        address of integer operand2)
local variable
    integer temp
begin
    temp = operand1
    operand1 = operand2
    operand2 = temp
end swap
```

```
function partition( start address of integer array,
                        integer start, end ) returns an integer
local variables
    integer pivotValue, pivotIndex, mid, scan

begin
    pivotIndex = start
    pivotValue = array[start]
    for  scan = start + 1 to end do
        if (array[scan] < pivotValue) then
            pivotIndex = pivotIndex + 1
            swap(array[pivotIndex], array[scan])
        end if
    end for
    swap(array[start], array[pivotIndex])
    return pivotIndex
end partition
```

You can download the MIPS simulator at: http://sourceforge.net/projects/spimsimulator/files/
Select the latest version of QtSpim for either Windows, MAC or Linux.

You should submit your homework via the Internet by following the directions at:
http://www.cs.uni.edu/~fienup/cs1410s19/homework/submissionDirections.html

You need to put the following files in a `hw7` folder and zip the folder to create a `hw7.zip` file. (On Windows you can a .zip file by right-clicking on the `hw7` folder and selecting `Send to|Compressed (zipped) folder`)

Your `hw7.zip` should contain the files:

- the MIPS assembly language program, e.g., hw7.s from any text-editor (e.g., WordPad)
- a window capture of the QtSpim simulator **after running** your assembly language program with the array values: `6 3 1 4 5 2 3 8 9 7 5`. **Make sure the sorted array are visible in the data section of the screen capture.** You can capture this window by (1) right-clicking anywhere in the window to make it the "currently active" window, (2) while holding down the <Alt> key, press the <PrtScn> key to capture the window into the Window's clipboard, and (3) open some word processor (Word, Open Office, etc.) and paste the image into the document. Add your name to this document before saving it.

On the top of the "directions" web-page, is a link to the submission tool (NOTE the https:// start of link) (https://www.cs.uni.edu/~schafer/submit/which_course.cgi). You'll need to enter your CatID username and password when requested.