COMP2011

Lab and Tutorial Nine (Total Marks: 100)

Deadline 23:59, 20 Nov. 2014

1. The partitionIt method is shown below.

```
public int partitionIt(int left, int right, long pivot)
   {
   int leftPtr = left - 1;           // right of first elem
   int rightPtr = right + 1;         // left of pivot
   while(true)
      {
      while(leftPtr < right &&       // find bigger item
            theArray[++leftPtr] < pivot)
         ; // (nop)
```

Breakpoint A

```
      while(rightPtr > left &&       // find smaller item
            theArray[--rightPtr] > pivot)
         ; // (nop)
```

Breakpoint B

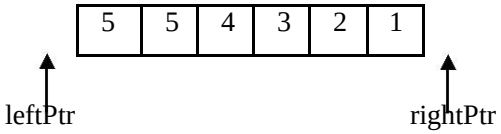
```
      if(leftPtr >= rightPtr)        // if pointers cross,
         break;                      //    partition done
      else                           // not crossed, so
         swap(leftPtr, rightPtr);    //    swap elements
```

Breakpoint C

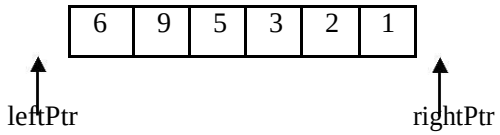
```
      } // end while(true)
   return leftPtr;                   // return partition
   } // end partitionIt()
```

Three breakpoints, namely, A, B, C, have been set in the program. For each of the following arrays and pivots, at each breakpoint, show the positions of leftPtr and rightPtr and the array content, respectively (the initial positions of leftPtr and rightPtr have been given).

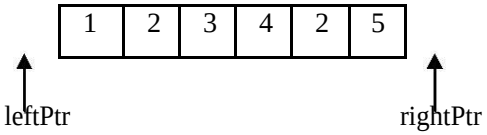
(1) Pivot: 5



(2) Pivot: 5



(3) Pivot: 5



2. Download the quickSort1.java file from the Blackboard. In this implementation of the quicksort algorithm, the rightmost array element is selected as the pivot in each partition. Moreover, when partitioning an input array, an array element that is equal to the pivot can be in either the left partition or right partition. Revise the program so:

   (1) the leftmost array element is selected as the pivot value in each partition;

   (2) when partitioning an input array, an array element that is equal to the pivot value can only be put into the left partition (the left partition contains all array elements smaller than or equal to the pivot, while the right partition contains all array elements bigger than the pivot);

   (3) print the pivot and display the array after each partition.

What to submit:

(1) The report that contains (a) the results of Question 1; (b) the output of your program from Question 2.

(2) The java source code of your program from Question 2.