Homework Turnin

Account: 6G_06 (rgalanos@fcps.edu)
Section: 6G
Course: TJHSST APCS 2016-17
Assignment: 09-01

Receipt ID: a2ac45105d3a346d840d3a87891de4f7

Turnin Successful!

The following file(s) were received:

HeapSort.java (2561 bytes)

```java
//Name:     Date:
import java.text.DecimalFormat;

public class HeapSort
{
   public static int SIZE;  //9 or 100

   public static void main(String[] args)
   {
   //Part 1: Given a heap, sort it. Do this part first.
   //       SIZE = 9;
   //       double heap[] = {-1,99,80,85,17,30,84,2,16,1};
   //       display(heap);
   //       sort(heap);
   //       display(heap);

   //  Part 2:  Generate 100 random numbers, make a heap, sort it.
      SIZE = 100;
      double[] heap = new double[SIZE + 1];
      heap = createRandom(heap);
      display(heap);
      makeHeap(heap, SIZE);
      display(heap);
      sort(heap);
      display(heap);
   }

   //******* Part 1 ******************************************
   public static void display(double[] array)
   {
      for(int k = 1; k < array.length; k++)
         System.out.print(array[k] + "    ");
      System.out.println("\n");
   }
   public static void sort(double[] array)
   {
      /* enter your code here */

      for(int i=array.length-1;i>1;i--)
      {
         swap(array,1,i);
         heapDown(array, 1, i-1);
      }

      if(array[1] > array[2])   //just an extra swap, if needed.
         swap(array, 1, 2);
   }
   public static void swap(double[] array, int a, int b)
   {
      double temp = array[a];
      array[a] = array[b];
      array[b] = temp;
   }
   public static void heapDown(double[] array, int k, int size)
   {
      int left = 2 * k;
      int right = 2 * k + 1;
      if(k > size || left > size)
         return;
      if(right > size)
      {
         if(array[k] < array[left])
            swap(array, k, left);
      }
      else
      {
         int maxChild = (array[left] > array[right])? left:right;
         if(array[k] < array[maxChild])
         {
            swap(array, k, maxChild);
            heapDown(array, maxChild, size);
         }
      }
   }

   // ****** Part 2 *******************************************

   //Generate 100 random numbers (between 1 and 100, formatted to 2 decimal places)
   public static double[] createRandom(double[] array)
   {
      array[0] = -1;   //because it will become a heap
      DecimalFormat df = new DecimalFormat("0.00");
      for(int i=1;i<=100;i++)
      {
         array[i] = Double.parseDouble(df.format(Math.random()*99+1));
      }
      return array;
   }
   //turn the random array into a heap
   public static void makeHeap(double[] array, int size)
   {
      for(int i = size/2;i>0;i--)
      {
         heapDown(array, i, size);
      }
   }
}


```