CS 146

Data Structures and Algorithms

Summer Semester 2015

Department of Computer Science
San José State University
Instructor: Ron Mak

Homework #4
Insertion Sort and Shellsort

Assigned:	Tuesday, July 7
Due:	Monday, July 13 at 11:59 pm
	100 points max

The purpose of this assignment is to give you a closer look at the **insertion sort** and **Shellsort** algorithms.

Write a program that will sort integer arrays of sizes 100 1000 10,000 and 100,000 using both sorting algorithms. Both algorithms should sort the same array, so make a copy of it each time before sorting.

The algorithms should sort, in the lowest to highest (increasing) order:

- An unsorted array of unique random numbers in random order.
- An array of unique numbers that's already sorted in increasing order. These numbers do not need to be random. Your algorithms should not know ahead of time that the array was already sorted.
- An array of unique numbers that's already sorted in reverse (decreasing) order. These numbers do not need to be random. Your algorithms need to resort the numbers in increasing order without knowing ahead of time that the array was already sorted in reverse order.
- An array of all zeroes. Your algorithms should not know ahead of time that all the array elements contain the same value.

Do Shellsort twice:

- First use the suboptimal sequence for h, which starts at half the length of the array and is halved for each pass.
- Then use Knuth's interval sequence 1, 4, 13, 40, ... (reversed) for values of h. The first value of h should be as large as possible but under half the array length.

Verify that your arrays are properly sorted!

Generalize your program enough to accommodate other sorting algorithms in the future.

Output

For each sort, you program should print:

- How much time it took in milliseconds.
- How many comparisons it made between two values.
- How many moves it made of the values.

  Examples:

  - If you shift five values over one position each, that's five moves.
  - Two values exchanging places (a swap) is two moves.

You can output these results in a single table.

Sample output (http://www.cs.sjsu.edu/~mak/CS146/assignments/4/output-4.txt):

```
                ===== Unsorted Random Unique =====

N = 100

                ALGORITHM          MOVES       COMPARES   MILLISECONDS
           Insertion sort          2,723          2,724              1
     Shellsort suboptimal            590            826              0
          Shellsort Knuth            627            718              0

N = 1,000

                ALGORITHM          MOVES       COMPARES   MILLISECONDS
           Insertion sort        247,291        247,293              9
     Shellsort suboptimal         11,857         15,339              1
          Shellsort Knuth         13,322         14,922              1

N = 10,000

                ALGORITHM          MOVES       COMPARES   MILLISECONDS
           Insertion sort     24,986,430     24,986,435             74
     Shellsort suboptimal        219,431        275,513             10
          Shellsort Knuth        206,966        228,206             10

N = 100,000

                ALGORITHM          MOVES       COMPARES   MILLISECONDS
           Insertion sort  2,502,386,612  2,502,386,607         10,227
     Shellsort suboptimal      3,718,361      4,394,436             55
          Shellsort Knuth      3,668,296      3,928,932             48

                ===== Sorted Unique =====


 • • •
```

Teamwork

You may work individually as a team of one, or you can partner with another student as a team of two.

You can be on only one team at a time. If you partner with someone, both of you will receive the same score for this assignment. You'll be able to choose a different partner or work alone for subsequent assignments.

What to turn in

Create a zip file containing:

- Your Java source files.
- Any instructions on how to build and run your code.
- Text files containing your outputs
- A short report (1 or 2 pages) that describes your conclusions from doing this assignment.

Name the zip file after yourself or yourselves.
Examples: **`smith.zip`**, **`smith-jones.zip`**

Each team should email the zip file to ron.mak@sjsu.edu. Your subject line must be:

`CS 146 Assignment #4` ***Your name(s)***

Example:

`CS 146 Assignment #4 Mary Smith & John Jones`

If you work with a partner, you should email only one assignment between the two of you. Whoever emails the assignment should CC the partner so that when I send you your team score, I can just do a "Reply all".