CS 284D: Homework Assignment 3
Due: April 21st, 11:55pm

1 Assignment Policies

Collaboration and IP Policy. Do homework individually. Do ask questions if anything is not clear. You can collaborate to understand the material. Do not solve the problem with others. Do not use any solution on the Internet. Do not exchange code with other students. **Any solution found online or in another submission will be penalized.**

Late Policy. No late submissions will be allowed without consent from the instructor. For urgent or unusual circumstances e-mail the instructor.

2 Assignment

2.1 Libray of Sort

Implement a generic sorting library that has

- **Input: array of objects of any generic type.**
- **Output: sorted array.**

3 Part I [10 Points]

A comparator that returns +1 if the first object argument is smaller than the second object, -1 if greater, and zero if equal.

```
1
public interface Comparable<Item>{
3   public int compareTo(Item that)
}
```

For example,

```
1 public class Date implements Comparable<Date>{
    ...
3 }
```

4 Part II [60 Points]

Implement the class of sorting algorithms:

4.1 Selection sort <10 Points>

```
public class Selection{
  public static void sort(Comparable [] a){
  ...
  }
}
```

4.2 Insertion sort <10 Points>

```
public class Insertion{
  public static void sort(Comparable [] a){
  ...
  }
}
```

4.3 Bubble sort <10 Points>

```
public class Bubble{
  public static void sort(Comparable [] a){
  ...
  }
}
```

4.4 Shuffling <10 Points>

```
public class Shuffling{
  public static void shuffling(Comparable [] a){
  ...
  }
}
```

4.5 Merge sort <10 Points>

```
public class Merge{
  public static void sort(Comparable [] a){
  ...
  }
}
```

4.6 Quick sort <10 Points>

```
public class Quick{
  public static void shuffling(Comparable [] a){
  ...
  }
  public static void sort(Comparable [] a){
  ...
  }
}
```

5 Test [30 Points]

5.1 Experiments <20 Points>

Implement the main method that calls examples to test the library for two test scenarios. Use the generic comparator interface and implement the object class of first names and dates for each scenario:

1. Given an array of first names, sort the array with each algorithm.
2. Given an array of dates e.g., 20200403, sort the array with each algorithm.

For each of above scenarios, perform below operations:

1. generate an array of objects.
2. sort the array with each of above six sorting algorithms.
3. compute the wallclock time and the number of swaps in sorting.
4. repeat operation 1-3 for 100 times and average the wallclock time and the number of swaps.

5.2 Analysis <10 Points>

Fill below table from your experiments.

	Wall clock time	**Number of swaps**	**Complexity class**
Selection sort			
Insertion sort			
Bubble sort			
Shuffling sort			
Merge sort			
Quick sort			

Submission Instructions:
Please submit a zip archive named hw4.zip, with the following files in it:
Comparable.java
Dates.java

Firstnames.java
Insertion.java
Selection.java
Bubble.java
Shuffling.java
Merge.java
Quick.java
MyTest.java
Table.txt

Please make sure your name and pledge are on all documents. No credit will be given for submissions with other file types.

IP Policy By receiving this homework as part of me taking the class CS284D I acknowledge that the content of this document will not be circulated as a whole or in part and in any implicit or explicit way. Anything else breaches the code of student conduct and, importantly, violates the federal laws of the US for intellectual property (cf. U.S. Copyright Law, December 2016). The instructor of this course maintains in its entirety the copyright of every document or electronic material (including slides).