Points: 100
Due Date: Monday, November 19, 2018 @ 11:59pm

For this assignment you are going to implement several sorting algorithms.

RESTRICTIONS:

- You may **NOT** import **java.util.Comparator**
  - you must write your own comparators
- You may **NOT** import **java.util.Arrays**
  - You must write your own copy methods for any arrays
  - You may **NOT** use any of the Java Array sorting features.
- You may **NOT** import any other Java container class.
  - e.g. you must use your own Queue class.

Task 1:

Create an employee Class that encapsulates the concept of an employee. The attributes of an employee are:

- id
  - a random integer in the range 0 to 99999999 (i.e. like a social security number)
  - we will ignore the fact that we may get duplicate id numbers
- name
  - a String of a random length between 5 and 10 characters (inclusive) made up of a random set of lower case characters
- dept
  - a random integer in the range 1 to 5 (inclusive)
- hired
  - a random integer in the range 2008 to 2018 (inclusive)

Task 2:

Create a class named Sort that will act as a container for the following generic array sorting algorithms:

- simpleBubbleSort
  - a brute force bubble sort that just uses a pair of nested loops
  - this needs to be a generic bubble sort
  - this needs to be a stable sort
- insertionSort
  - the insertion sort as discussed in class
  - you may use the code from the Java Illuminated text modified to be generic
  - make sure it is a stable sort
- selectionSort
  - the insertion sort as discussed in class
  - you may use the code from the Java Illuminated text modified to be generic
  - make sure it is a stable sort
- mergeSort
  - this should be the recursive mergeSort described in the textbook
- quickSort
  - this should be the recursive quickSort described in the textbook
  - you may have to modify this code
- radixSort
  - this should be a generic sort
  - the radixSort should be able to support between two and four keys

- the first parameter in the parameter list should be the array being sorted.
- the remaining parameters in the parameter list should be the keys, ordered left to right from most significant to least significant
- you should use the Radix sort described in class and not the bucket approach described in the textbook

Task 3:

- Create a client class that
  - Generates an array of 100,000 employees
  - Sort the employee array on name using the merge sort
  - Sort the employee array on deptment using the quick sort
  - Sort the employee array on id using the bubble sort
  - Sort the employee array on name using the insertion sort
  - Sort the employee array on id using the selection sort
  - Sort the employee array using the radix sort so that
    - All employees are sorted by department
    - Within a department grouping all the employees are sorted by hire date
    - Within a department and hire date grouping all the employees are sorted by their name
- Since the list of employees is long
  - You will not print out the unsorted or sorted employee lists, instead,
  - Print out the time that it takes to run each sort
  - Suggestion:
    - Make a test run of 100 employees and inspect the results to make sure that they are correctly ordered but you should not display them in your Word document

- Caution
    - Make sure that you are passing the same unsorted list to each of your sort routines.
    - If you follow the textbook code the container that is passed in as a parameter is the container that is sorted.

Turning in your assignment:

- **Make sure that all of your code is properly documented.**
- Turn in your assignment using the standard method.
- Create a Word document and copy and paste each of your Java files into the document.
    - You only need to include the Java files specific to this assignment
        - Comparator relations files
        - Sort class
        - Client class
        - Any other files written specifically for this assignment.
- Paste the screenshots showing the complete output of a complete run of your program after the Java code in your document.
- Export your NetBeans project to a zip archive.
- Turn in the Word document and zipped project as to separate files in a single Blackboard submission.