Object-Oriented Programming II (CS 116 - 03)
Lab 4
Due: 02/10/2020 at 3:00pm

Goals for today

Searching and Sorting

Objectives

- 15 Points - SortSearch
- 10 Points - Student Grades

Description

SortSearch

Write `SortSearch` class that contains the following methods:

- `selectionSortA(int [] A)` : Sort the array `A` into increasing order using `selection sort algorithm`.
- `selectionSortD(int [] A)` : Sort the array `A` into decreasing order using `selection sort algorithm`.
- `insertionSortA(int [] A)` : Sort the array `A` into increasing order using `insertion sort algorithm`
- `insertionSortD(int [] A)` : Sort the array `A` into decreasing order using `insertion sort algorithm`
- `BubbleSort(int [] A)` : Sort the array `A` into increasing order using `bubble sort algorithm`. The `bubble sort algorithm` is comparison-based algorithm in which each pair of adjacent elements is compared and the elements are swapped if they are not in order.

```
While the array is not sorted
  For each adjacent pair of elements
    If the pair is not sorted
        Swap its elements.
```

- `String[] sortArrayOfStrings(String[] array)` : Sort the array `array` into increasing order using the `insertion sort algorithm`.
- `shuffle(int [] A)` : Rearrange the items in the array `A` into a random order. Putting the elements of an array into a random order is much less common problem, but is a bit more fun. The typical case of this problem is shuffling a deck of cards. A good algorithm for shuffling is similar to selection sort, except that instead of moving the biggest item to the end of the list, an item is selected at random and moved to the end of the list.
- `int find(int[] A, int N)` : Search the array `A` for the integer `N`. If `N` is not in the array, then `-1` is returned. If `N` is in the array, then the return value is the first integer `i` that satisfies `A[i] == N`.
- `int binarySearch(int[] A, int N)` : Search the array `A` for the integer `N`. `A` must be sorted into increasing order. If `N` is in the array, then the return value, `i`. If `N` is not in the array, then the return value is `-1`.

Write a client class to test all your methods.

Student Grades[1]

Write `StdentGrades` class that encapsulating the concept of student grades on a test, assuming student grades are composed of a list of integers between `0` and `100`.
Write the following methods:

- A `constructor` with just one parameter, the number of students; all grades will be randomly generated.
- `Accessor`, `mutator`, `toString` and `equals` methods
- A method returning an array of the grades sorted in `ascending order`.
- A method returning an array of the grades sorted in `descending order`.
- A method returning the highest grade.
- A method returning the average grade.
- A method returning the median grade (*Hint: the median grade will be located in the middle of the sorted array of grades*)
- A method returning the mode (the grade that occurs the most often) (*Hint: create an array of counters; count how many times each grade occurs; then pick the maximum in the array of counters; the array index is the mode.*

Write a client class to test all your methods.

[1]Credit: Java Illuminated 5th edition