

CS 4306: Algorithms  
Spring 2022  
Assignment 4 – Chapter 4 – 100 points

**Note 1:** If you re-upload revised files, you must re-upload ALL files as the system keeps the most recent uploaded submission only. No zip files, just individual Java files.

**Note 2:** In addition to the *Algorithm Design Block* (see below if required), please include comments for the code implementation of the algorithm when applicable.

The goal of this assignment is to get more insight into the performance of **Interpolation Search** on sorted lists. Interpolation search is a decrease-and-conquer based algorithm.

Develop this assignment in incremental fashion.

**Part 1:** Develop **class InterpolationSearch()** that implements interpolation search discussed in chapter 4. The class design must include three variables:

Found: boolean variable to indicate if key value is found in the list or not (T/F).

Index: integer variable that holds the index of target value if found in the list. Otherwise it holds -1.

Divisions: Integer variable to hold number of divisions performed to find a value (or exhaust the search).

Implement the class constructor such that it takes an array and key value to search for. Before you proceed to next part, please test this part on its own with simple data to make sure it works properly and those three variables are populated correctly.

**Part 2:** Implement **class TestIS()** to define at least the following methods:

Method **RandomDistinct()**: A method to generate array (call it Values[ ]) of size 1024 and containing random and distinct numbers between 1 and 9999. The method then sorts the array Values[ ] in ascending order.

Method **RunIS()**: A method to generate a table that looks as follows (populated with dummy data to show expected format). Allow the user to enter the table size (see menu, option 1):

Key	Found	Index	Divisions
1554	True	476	3
54	True	211	2
7465	False	-1	4
3261	True	431	3
9827	True	12	1
543	True	379	2
6683	True	92	3
1	False	-1	4
2723	True	326	5
329	True	288	2
1554	True	476	3
. . .			
Divisions average:	<computer average value of all divisions>		
Difference:	<3.322 - computed average>		

Notice that 3.322 is  $\log_2(\log_2 1024)$ . 1024 is our array size.

To generate the table, use the table size input value (from the user) and run a loop such that each iteration it

generates a key value (between 1 and 9999),  
creates an object of class **InterpolationSearch()** that takes array Values[ ] and Key, and  
adds object data to the output table (as shown above).

Integrate the following simple menu in the program with these options:

```
-----MAIN MENU-----
1. Create and display array Values[]
2. Read output table size
3. Run algorithm and display outputs
4. Exit program
```

Enter option number:

Notice that array `Values[]` is created once and does not change while running options 2 and 3. It changes only when option 1 is selected.

Option 1 populates `Values[]` with random and distinct numbers (between 1 and 9999), and then displays the array on screen in rows and columns (30 values per row), use tabs to separate values:

```
1    15    35    44    68    87    89    97    101   116   119   120   126   124   130   134   138   139   142   147   152   158   163   165
169  170   171   174   178   183   199   200   221   225   275   279   280   285   290   295   297   299   301   308   311   312   317   323
327  340   345   352   357   359   260   267   370   387   390   392   . . .
```

### Algorithm Design Block:

No algorithm design is required for this assignment. However, make sure you document your code properly.

### Submission:

Do not forget to include author header in each submitted file, as shown below, and do not forget to document your code. **No author header or no proper documentation, no points!**

```
// Name:          <your name>
// Class:         CS 4306/1
// Term:          Spring 2022
// Instructor:    Dr. Haddad
// Assignment:    4
```

Please submit your .java files `InterpolationSearch()` and `TestIS()` to the assignment submission folder in D2L by the due date posted in D2L. Make sure that your code is running correctly right before you upload your files. **No zip files or late submissions are accepted.**