**Course: Advanced Studies-1  
Code: CS401  
Name: Akshitha Bedre Shivakumar**

**A20544641**

**Final Project:**

The project requirements are for developing a Java program that compares the difficulties of sorting and searching methods. We are using the Java development kit (JDK) which is a developer’s tool on our desktop which uses JVM along with eclipse IDE tool to run the program. We are creating a menu to select each sorting and searching operations. Once the list is produced, the user can choose between two sorting algorithms: one basic (selection sort, insertion sort, bubble sort) and one O(Nlog2N) sort (Quick sort, Merge Sort, Heap sort Comparing search methods works in the same way: perform both a linear search with the original list and a BST with the sorted list. We are creating a hash function list using the same data and compare the complexity to linear, binary, and hash function searches. And we are discussing the outcome of our program with the Big-O theory of the algorithm you chose.

* The Java software offered looks to be centered on sorting and searching methods. The primary application, CS401prj, is built around a menu-driven user interface that allows the user to select between several sorting and searching options.
* Linear search method searches the given array for the specified element using a linear search. It iterates across the array, using the CS401prj.compare method to compare each element to the target element. If a match is discovered, the index, element, and number of comparisons conducted during the search are printed. If the element is not found, a message is printed along with the total number of comparisons performed.
* Binary search method does a binary search for the supplied element in a binary search tree (BST). It generates a new BinarySearchTree, builds a sorted BST from the data, and then does the binary search. It displays the outcome (whether the element was found or not) as well as the number of comparisons performed throughout the search.
* The Hash method searches for hash functions with a custom HMap (hash map) implementation. It populates the hash map with array elements before utilizing the contains method to see if the provided element is present in the hash map.
* Sorting class contains implementations of two sorting algorithms: selection sort and heap sort. Let's go through each method: The selection sort algorithm is implemented by this method it loops through the array, finding the smallest element in the unsorted section and exchanging it with the initial unsorted element each time. The number of comparisons done when sorting is tracked and returned. The heap method implements the heap sort algorithm. It first builds a max heap from the array by calling the reheapDown method. Then, it repeatedly swaps the root (maximum element) with the last element in the heap and restores the heap property using reheapDown.

**Working:**

* It initializes variables such as user input, data type, and search-related variables, and to populate the array from a file, we are using different test cases with 1000, 2000 and 3000 data entry files.
* The program then enters a while loop, displaying a menu from which the user can select a sorting or searching action from Selection Sort, Heap Sort, Linear Search, Binary Search Tree, Hash Function Search, and Exit are all options.
* For the selected option, user input is accepted and sorting or searching activities are carried out based on the option selected.
* If the user selects Selection Sort or Heap Sort, the associated algorithm is applied to an array clone and the sorted array is printed, along with the number of comparisons performed throughout the sorting process.
* The user is requested to enter the element to search for while using Linear Search. The index and the number of comparisons are shown and the user is requested to enter the element to search for while using Binary Search Tree Search. The number of comparisons and if the element was discovered are displayed.
* The user is requested to enter the element to search for while using Hash Function Search. The Searching class's hash Search function is invoked.
* We can at any time chose to exit the current choices by selecting exit option.
* To run the program use **project.jar** file into the computer.
* Open terminal and enter in the command line : **java -jar project.jar.**
* Give the user input from the menu requirements.