CS 201, Spring 2018

Homework Assignment 2

Due: 23:59, April 16, 2018

In this homework, you will study the search problem. That is, given an ordered collection of items (i.e., a sorted array of integers) and an item of the same type to be searched (key) in this collection, the problem is to find out if the key exists in the collection or not. If it exists, you are supposed to return the position of the key (i.e., the index). Two alternative algorithms that solve this problem are discussed below (also in class). Each algorithm has a different time complexity. The goal of this homework is to evaluate the growth rates of both algorithms using different inputs.

Algorithm 1: Linear search which works in O(N) time, where N is the size of the collection.

Algorithm 2: Binary search which works in O(logN) time, where N is the size of the collection.

Note that Binary Search works on only sorted collections, so assume the inputs are always sorted in ascending order for both algorithms.

ASSIGNMENT:

1. Study the algorithms and understand how the upper bounds are found for the running time of each solution.

2. Use the implementations given in the slides and write a driver (main) function that calls these functions. Then, create sorted arrays of different sizes. You are expected to try many different input sizes, both small inputs and very large inputs (as large as around 1 000 000 000). For each input size, consider the following positions for searched keys: When the key is (i) close to the beginning, (ii) around the middle, (iii) close to the end, and (iv) not existent in the collection. Run each array size and key position combination and record the execution times. **Do not include the time elapsed to allocate the array and initialize its entries.**

3. Use these results to generate a plot of running time (y-axis) versus the input size *n* (x-axis), per key position (i.e., for (i), (ii), (iii), and (iv)). Specifically, you are expected to produce plots similar to Figure 2.3 of the handout chapter on algorithm analysis.

4. Based on your plots indicate the best, average and worst cases for each algorithm.

5. Provide the specifications (processor, RAM, operating system etc.) of the computer you used to obtain these execution times. **You can use any computer with any operating system for this assignment**.

6. Also, plot the expected growth rates obtained from the theoretical analysis (as given for each algorithm above) by using the same *n* values that you used in your simulations.

7. Finally, compare the expected growth rates and the obtained worst case results, and discuss your observations in a paragraph.

You can use the following code segment to compute the execution time of a code block. For these operations, you must include the ctime header file.

```
//Store the starting time
double duration;
clock_t startTime = clock();

//Code block
//...

//Compute the number of seconds that passed since the starting time
duration = 1000 * double( clock() - startTime ) / CLOCKS_PER_SEC;
cout << "Execution took " << duration << " milliseconds." << endl;
```

An alternative code segment to compute the execution time is as follows. For these operations, you must include the chrono header file.

```
//Declare necessary variables
std::chrono::time_point< std::chrono::system_clock > startTime;
std::chrono::duration< double, milli > elapsedTime;

//Store the starting time
startTime = std::chrono::system_clock::now();

//Code block
...

//Compute the number of seconds that passed since the starting time
elapsedTime = std::chrono::system_clock::now() - startTime;
cout << "Execution took " << elapsedTime.count() << " milliseconds." << endl;
```

NOTES:

1. This assignment is due by 23:59 on April 16, 2018.

2. This homework will be graded by your TA Gozde Gunesli (nur.gunesli@bilkent.edu.tr). Thus, you may ask your homework related questions directly to her.

3. Before the deadline, you should email your homework to nur.gunesli@bilkent.edu.tr with subject line "**CS 201 - HW2**". No hardcopy submission is needed. The standard rules about late homework submissions apply. Please see the course syllabus for further discussion of the late homework policy as well as academic integrity.

4. In this assignment, you must submit a report (as a pdf file) that contains all information requested above (plots, computer specification, discussion) and a cpp file that contains the main function that you used as the driver in your simulations in a single archive file. The name of this archive file should be: HW2_secX_Firstname_Lastname_StudentID.zip where X is your section number. (you do not need to submit the solution functions).