CS 201, Fall 2019

Homework Assignment 2

Due: 23:55, December 2, 2019

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, your solution needs to return the position of the key (i.e., the index). If not, it needs to return -1.

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(*log N*) time, where *N* is the size of the collection.

Note that the binary search algorithm 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 four possibilities for searching a key: (i) the key is close to the beginning, (ii) the key is around the middle, (iii) the key is close to the end, and (iv) the key does not exist in the collection. Run your program for each array size and for each possibility and record the execution times. **Do not include the time elapsed to allocate the array and initialize its entries.**

3. Report the results in a table. Then, use these results to generate a plot of running time (y-axis) versus the input size *N* (x-axis), for each of the four possibilities (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. 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. Likewise, give these plots for each of the four possibilities (i.e., for (i), (ii), (iii), and (iv)).

7. 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 23:55 on Monday, December 2, 2019. You should upload your homework to the upload link on Moodle before the deadline. This upload link will be available between November 22 and December 5. No hardcopy submission is needed. The standard rules about late homework submissions apply. Please see the course web page for further discussion of the late homework policy as well as **academic integrity**.
2. You must submit a report (as a pdf file) that contains all information requested above (plots, tables, computer specification, discussion) and a cpp file that contains the main function that you used as the driver in your simulations as well as the solution functions in a single archive file.
3. You should prepare your report (plots, tables, computer specification, discussion) using a word processor (in other words, do not submit images of handwritten answers) and convert it to a pdf file. If you have any images of handwritten answers in your report, you will lose a considerable number of points.
4. The code segments given above may display 0 milliseconds when the value of *N* is small. Of course, the running time cannot be 0 but it seems to be 0 because of the precision of the used clock. However, we will not accept 0 as an answer. Thus, to obtain a running time greater than 0, please consider running an algorithm *M* times using a loop, and then divide the running time (which will be greater than 0) to *M*.
5. This assignment will be graded by your TA Gözde Nur Güneşli (nur.gunesli@bilkent.edu.tr). Thus, you may ask your homework related questions directly to her.