Faculty of Computer Science, IBA

Data Structures (3+1) Instructor: Ali Raza

Algorithm Complexity

Objective

The objective of this lab is to understand how we can analyze the time complexity of an algorithm. Develop searching algorithms and find the time complexity to analyze the growth rate of an algorithm as the data size gets larger.

Task

1. Sequential Search

This algorithm takes about N steps for an unsuccessful search and about N/2 steps on the average for a successful search.

2. Binary search

Divide the array into two parts; determine which of the two parts the key being sought belongs to then concentrate on that part and continue doing the same until no more division is possible. The total number of statements executed for N data size is approximately equal to $\log_2 N$.

Problem 1

Write a program that empirically tests the *time complexity* of the binary search and linear search algorithm. Create an array of integers and remove the duplicate values before to run the search algorithms to search a key value in an array.
Find time taken by these algorithms. To measure the system time use System.nanoTime();

Due to the fact that our algorithms are dependent on system resources therefore the time taken by same algorithm for same value might be different on different time slot. Therefore, system time is not consistent. Thus, rather than using the system time, count the number of comparison operations based on each algorithm in the array of different sizes. Compare the algorithms based on their growth rate as data size increases.

N	BinarySearch Comparison Statements	LinearSearch Comparison Statements
10	3	10
100	7	100
1000	--	--
5000	--	--

To verify your counts for the binary search, calculate log_2(N). Function to calculate log_2(N) is, Math.log(N)/Math.log(2). Draw graph in excel for the table shown above.

Problem 2

Write a function, **public void myfunc(int[] A, int N),** whose formula for the running time should be $T(N) = N^3 + N + N * \log(N)$. Feel free to use loops, conditional statements, etc. Also, compute the counts of primitive operations for

i) T(N)
ii) and for the dominant term in T(N)

for the values of N from 10 to 1000. Plot the counts for i) and ii) on a chart in MS Excel. At what value of N, the dominant term accounts for more than 99.5% of the counts? Submit the code and the MS Excel files.