**ALGORITHMS & COMPLEXITY**

**CS203.3**

**Dr. Rasika Ranaweera**

**Lab Exercise I**

**2021 Summer**

**Type:** Individual & Mandatory

**Duration:** 2 Hours (+ Homework)

**Lab Exercise (Algorithm Revision)**

**Important:**

* Objective is to understand, implement, and analyze familiar algorithms
* Bring your answer sheets (on top write your name, ID, course code) to the next class
* Learn from others or Google but DO NOT COPY

**Introduction**

Searching algorithms aim to find position of a target value within an array/list. Selection, merge, linear, binary, jump, ternary search are examples. Lets discuss about two of them.

**Linear Search**

Checks each element of the list until a match is found or reaches end

**Binary search**

Compares the target value to the middle element within a sorted array and eliminates irrelevant half

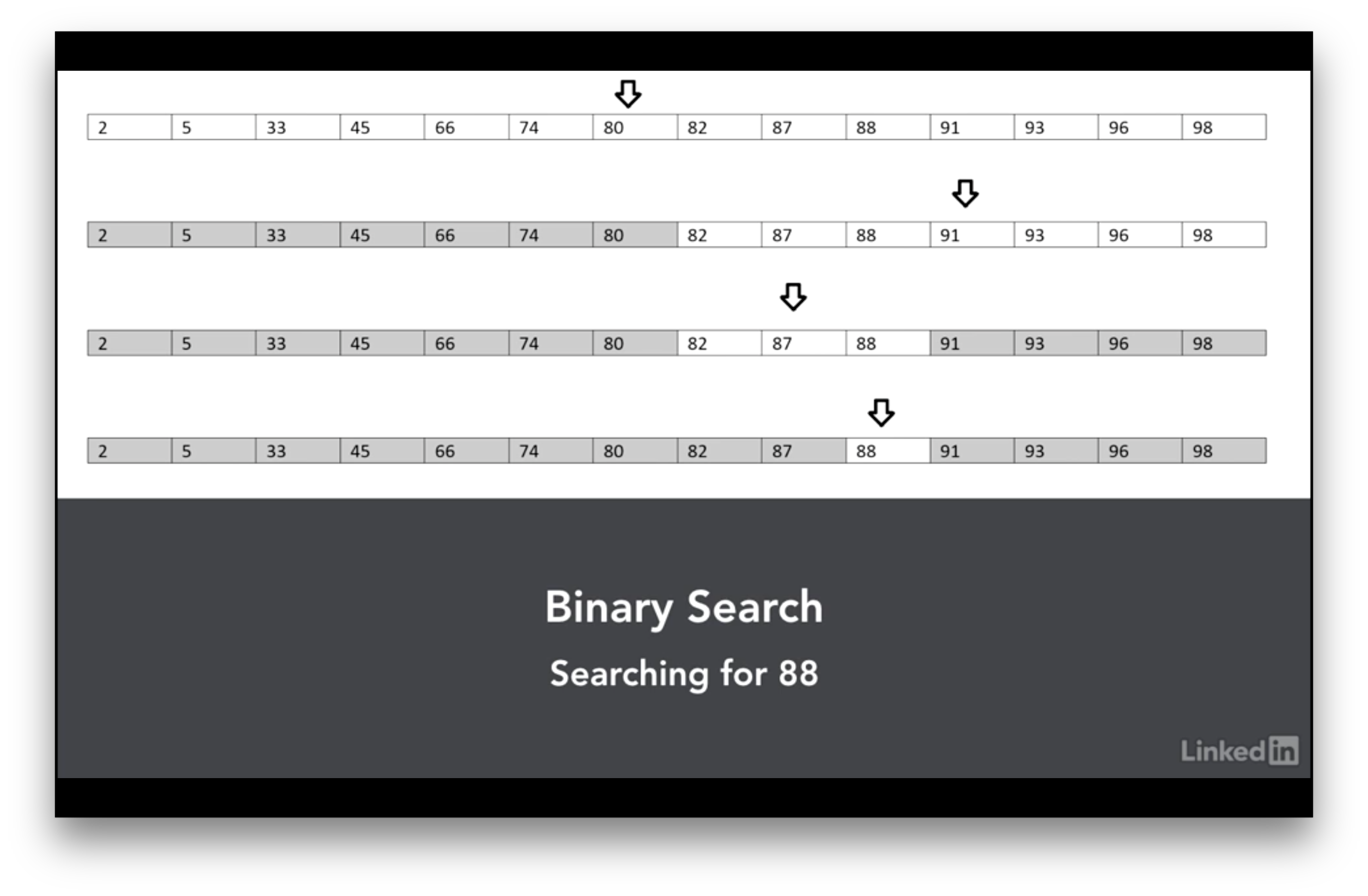


Image curtsey: <https://www.lynda.com/C-tutorials/Binary-search-explained/604241/636317-4.html>

**Assignment:**

Prepare a report answering the basic questions [R], then write programs [S] using any programing language, and finally plot the data on an excel file [E].

1. Write pseudo-codes for the above searching algorithms (you must also include references if used) [S].
2. Implement both algorithms using any program language [S]
   1. Write a program to generate any number of random integers in 0 to 100 range. Your program should get the size as a parameter and return the numbers as an array [S].
   2. Now implement the linear search. Pass the random array of 100 items as the list and “50” as the item to find [S].
   3. Compute the time to find 50 in the array and record the time consumed [S].
   4. Conduct this test for 10 times and plot the timing on a graph [E].
   5. Now conduct the same test for 1K (1000), 5K, 10K, … , 50K and plot the average timing [E].
3. Redo the same test for binary search.
4. Plot your finding on a graph (Red: Binary, Blue: Linear) and a table (including your computer specs) [E]

**References:**

<http://python-textbok.readthedocs.io/en/1.0/Sorting_and_Searching_Algorithms.html>