**Course Project**

**DeVry University**

**College of Engineering and Information Sciences**

**Course Number: CEIS295**

# Module 5: Searching Algorithms

# Objectives

* To check the speed of a linear search on a large dataset
* To compare the speed of a binary search on a large dataset when the data is sorted

Last week, we discovered sorting techniques that are very fast on large datasets. Why is it important to sort your data? Let’s compare the search speed of an unsorted dataset against the search speed against a sorted dataset.

If the data is not sorted, you will need to use a linear search. That means that you need to start with the first record. Then, you need to compare each record, one at a time, until you find the record that you are searching to find. When you step record-by-record, the search speed is O(n), which is considered “slow”.

On the other hand, if the data is sorted, you can use a binary search. In the binary search, you compare the middle record with the record to find. If the middle record is not the one, then you determine if it is bigger or smaller than the record to find. If the middle record is bigger, then the record to find must be in the lower half of the dataset. In one step, you have eliminated half of the records! You keep repeating this process where you eliminate half of the records on each step until you find your record or determine that it does not exist in the data. When you cut the data in half on each step, the search speed is O(log n), which is considered “fast”.

Let’s compare the search speeds with real-world data.

# Steps

1. Create a new folder in your CEIS295 folder called “Week 5 Project”. Create an Excel table in this folder so we can record the time that it takes to perform searching algorithms on real-world data. We are going to compare real-life speeds for the following searching algorithms:  
   1. Linear Search
   2. Binary Search

Your Excel table should look something like this:  
  


1. Download the ClientData100.csv file, ClientData1000.csv file, ClientData10000.csv file, and ClientData100000.csv file and place the files in your “Week 5 Project” folder. We will read the data in these files so we can work with real-world sized datasets to test our algorithms.
2. In this same folder, copy your Client class from last week that is based on the following UML diagram.
   1. The \_\_lt\_\_ method means “less than” and it should return True if this object is less than the other object (parameter). **Update the class and use the client id to determine if this object is less than the other object.**
   2. The \_\_eq\_\_ method means “equal to” and it should return True if this object is the same as the other object (parameter). Otherwise, it should return False. **Update the class and use the client id to determine if the objects are the same.**
   3. **Update the \_\_str\_\_ method so it returns \_\_client\_id\_\_, \_\_last\_name and \_\_first\_name in this format: 100001, Black, Jack**
   4. Also, make sure your name and the current date are listed at the top of the code.  
        
      
3. Download the files called LinearSearch.py, BinarySearch.py, and Quicksort.py. Place these code files in your “Week 5 Project” folder. We are going to use the Quicksort algorithm before we run our binary search.
4. First, let’s test the speed of the Linear Search algorithm. In this same folder, create a file called SearchingActualSpeeds.py. Type your name and the current date at the top of the code. Then, import the LinearSearch class, the BinarySearch class, the Quicksort class, the Client class, the date module from the datetime library, the random module, and the time module.
5. In the same SearchingActualSpeeds.py file, display your name and the current date in the output to show that you are the author of this code.
6. In the same SortingActualSpeeds.py file, put these lines next in your code. This way, you can uncomment the line with the name of the file that you want to test by removing the hashtag.  
     
   #input\_file\_name = 'ClientData100.csv'  
   #input\_file\_name = 'ClientData1000.csv'  
   #input\_file\_name = 'ClientData10000.csv'  
   #input\_file\_name = 'ClientData100000.csv'
7. Read the records from the ClientData100.csv records into Client objects and place the Client objects into a list called clients. Now, check to see how long it takes the Linear Search algorithm to search for 1000 random clients in the list. Type this time value into your Excel table.
8. Then, use the ClientData1000.csv file and check to see how long it takes to search for 1000 random clients when you have 1000 records in your dataset. Use the ClientData10000.csv file to check to see how long it takes to search for 1000 random clients when you have 10000 records in your dataset. Finally, use the ClientData100000.csv file to check to see how long it takes to search for 1000 random clients when you have 100000 records in your dataset. Write these times into your Excel table.
9. Test the speed for the Binary Search algorithm. Put the time that it takes to search for 1000 records using the different datasets in the Excel table. You will discover why the Linear Search is considered “slow” and the Binary Search is considered “fast”!

# Deliverables Part 5

* Complete the Module 5 Course Project Presentation deliverable
  + Client.py code – did you update \_\_lt\_\_, \_\_eq\_\_, and \_\_str\_\_?
  + SearchingActualSpeed.py code
  + Screenshot showing the code running with your name and date in the output
  + Excel “Table of Searching Speeds” table