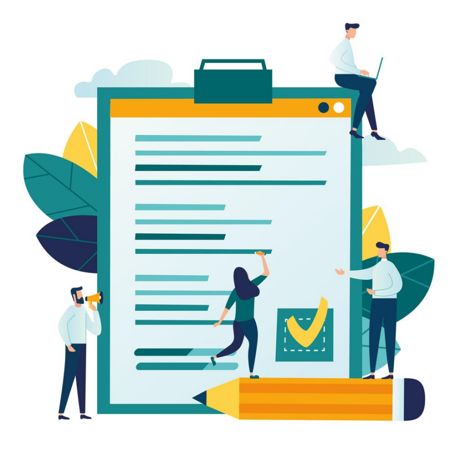
Full Stack Software Development

Course: Data Structures and Algorithm

Lecture On: Searching Algorithms

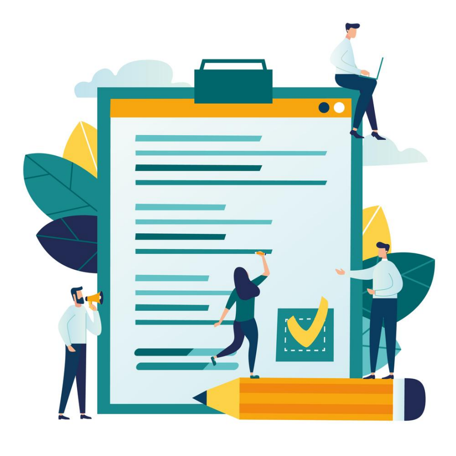
In the previous class, we covered….

- Arrays



Today’s Agenda

1 Searching Algorithms



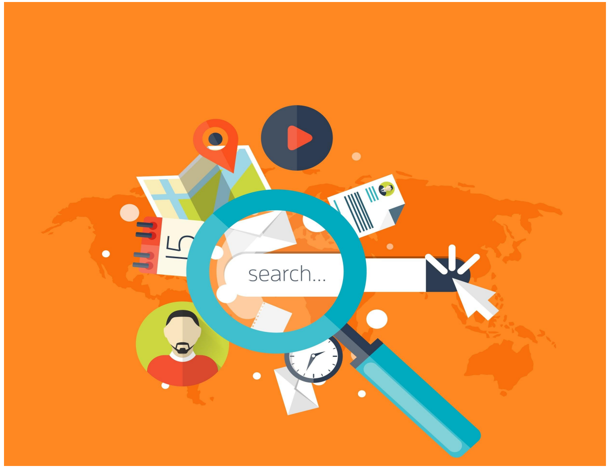
Searching

upGrad

Libraries have countless books, music stores have several CDs, Google has massive data, Amazon has innumerable things to sell and Myntra has a huge catalogue you can choose from!

Let's say I want to buy a Lego set from Amazon. How would I find the link to the relevant page? I'll probably search it!

Searching is a significant capability that one must be able to perform in this data-driven world!



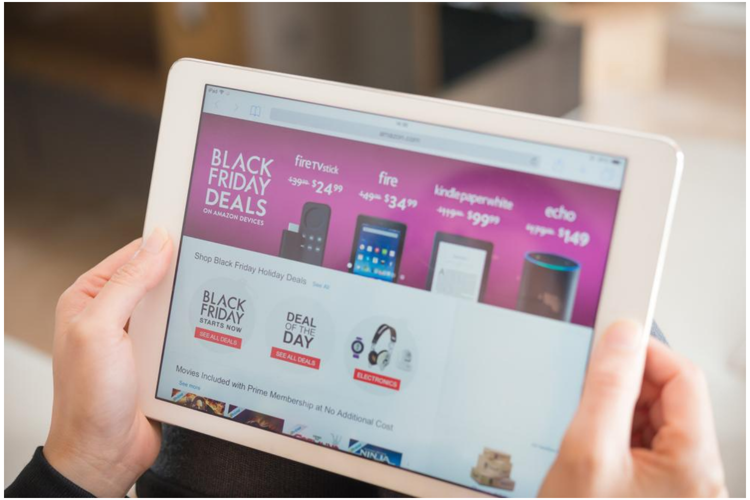
Searching

But given the huge amount of data that exists out there, ‘searching’ is not always a trivial task.

In a bit more formal sense, a Searching problem requires us to locate an element stored in a Data Structure. Let's learn more about searching for an element in linear data structures.

What happens when you search for a product on Amazon?

Searching for a product on Amazon is looking for it in a database, i.e., on persistent storage network. However, this problem can be broken down by finding its position, where the product details are stored on a data structure.



Searching

What do you think we mean by ‘**keys**’ while searching?

What will be the input? The data structure and the ‘key’ (or the data element) to be searched for.

What will be the output? The index of the data structure where the ‘key’ is located If the ‘key’ isn’t found, we would need to output accordingly.

As always, we will need to be as efficient as possible while Searching in terms of time and space that we use.

Searching

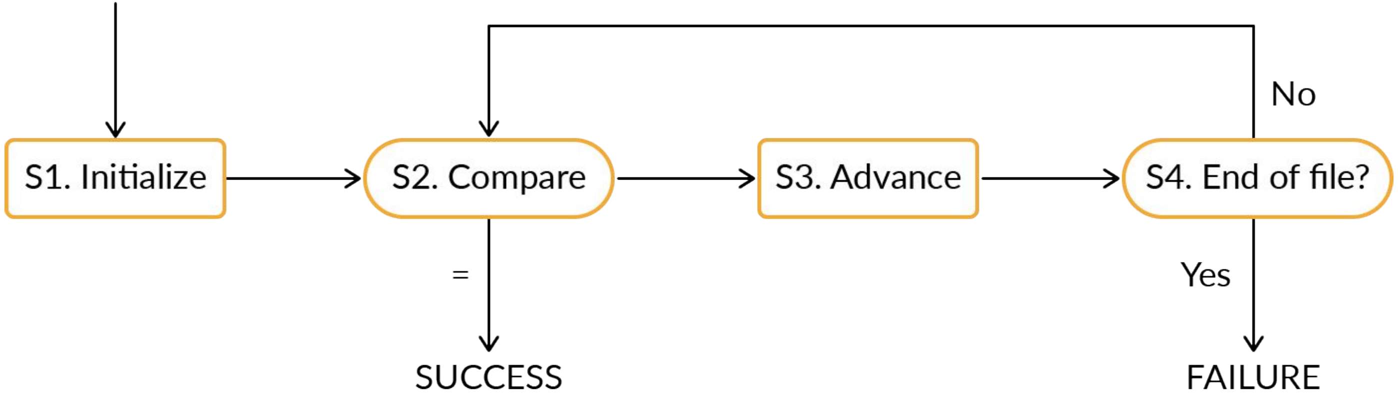
In this course, we will be looking at the following search algorithms:

- **Linear Search**
- **Binary Search**

upGrad

Linear Search

Do you remember how we traversed an array to look for an element in it? Inadvertently, we ended up performing Linear Search. Linear Search, or Sequential Search, as it is aptly called, entails searching through the entire linear data structure, one element at a time, until we find the 'key' or exhaust the entire data structure.

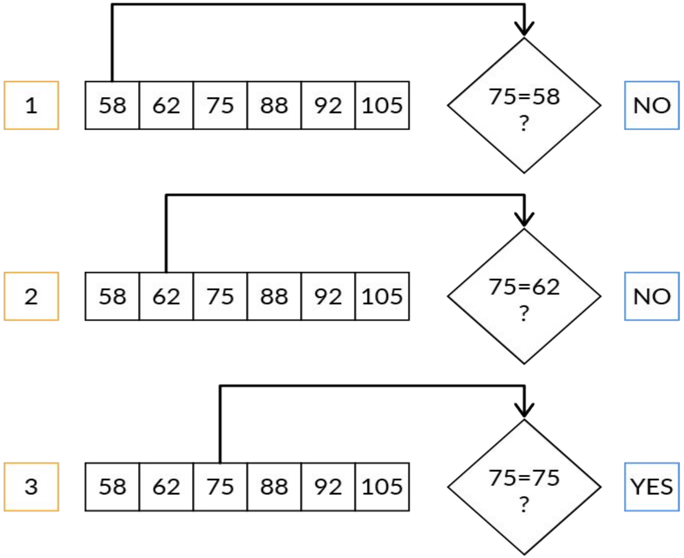


Linear Search

upGrad

List of elements : 58, 62, 75, 88, 92, 105

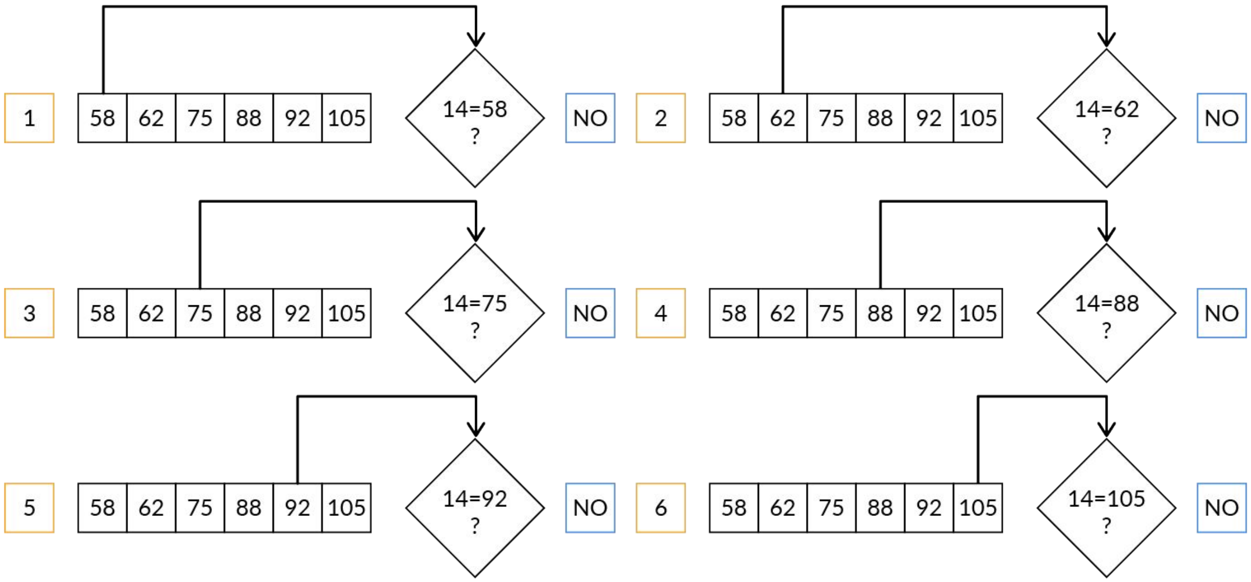
Elements to be searched: 75



Linear Search

List of elements : 58, 62, 75, 88, 92, 105

Elements to be searched: 14



Poll 2 (15 Sec.)

What is the worst time complexity of linear search?

1. O(n)
2. O(1)
3. O(logn)
4. None of these

Poll 2 (Answer)

What is the worst time complexity of linear search?

1. **O(n)**
2. O(1)
3. O(logn)
4. none

Poll 3 (15 Sec.)

What is the best time complexity of linear search?

1. O(n)
2. O(1)
3. O(logn)
4. None of these

Poll 3 (Answer)

What is the best time complexity of linear search?

1. O(n)
2. **O(1)**
3. O(logn)
4. None of these

Binary Search

Suppose we want to know what the word ‘weftage’ means.

Earlier, without knowing how to search we would have started with the first page of the dictionary and tried to match the word ‘weftage’ with all the other words in the dictionary one by one.

How long do you think this would take us? (The Oxford dictionary has over 1,80,000 words)

Binary Search

Let's explore better ways at performing this search rather than scanning the entire data set. Binary Search provides a much faster algorithm and guarantees better performance than Linear Search.

What would you do if you had to search for a word in the dictionary yourself? You would start with a more efficient method for starters, right?

Which component of the dictionary enables us to to do this?

A dictionary is sorted in alphabetical order!

Binary Search

Let’s look at how Binary Search actually works.

The way we opened the dictionary to see which word we arrived at, we first check the **middle** element of the array to look for the ‘key’. If it is found, we return the index. If it is not found, we can now know which **half** of the array might contain the ‘key’ that we are searching for, by comparing the ‘key’ and the data element we just checked.

Now, we proceed to perform Binary Search on the remaining half until we find the data element that we are looking for.

Poll 4 (15 Sec.)

What is the best case time complexity of binary search?

1. O(n)
2. O(logn)
3. O(1)
4. None of these

upGrad

Poll 4 (Answer)

What is the best case time complexity of binary search?

1. O(n)
2. O(logn)
3. **O(1)**
4. None of these

Poll 5 (15 Sec.)

What is the worst case time complexity of binary search?

1. O(n)
2. O(logn)
3. O(1)
4. None of these

Poll 5 (Answer)

What is the worst case time complexity of binary search?

1. O(n)
2. **O(logn)**
3. O(1)
4. None of these

Poll 6 (15 Sec.)

Binary search can be applied on any array.

1. True
2. False

Poll 6 (Answer)

Binary search can be applied on any array.

1. True
2. **False**
   (It can be applied to only ‘sorted’ arrays)

upGrad

Poll 7 (15 Sec.)

Time complexity of binary search:

1. Is different in different languages
2. Is different in different platforms
3. Both 1 and 2
4. Neither 1 nor 2

Poll 7 (Answer)

Time complexity of binary search:

1. Is different in different languages
2. Is different in different platforms
3. Both 1 and 2
4. **Neither 1 nor 2**

Coding Question

Find an element in a sorted array using:

- Iterative approach

Now perform binary search on array which we took for linear search and find the number of comparisons to find an element.

Poll 8 (15 Sec.)

Which is the smallest missing element in the given array: 13 14 15 16 17 18 29 30 31 32 33, having lower range as 12 and upper range as 35

1. 13
2. 12
3. 19
4. 35

Poll 8 (Answer)

Which is the smallest missing element in the given array: 13 14 15 16 17 18 29 30 31 32 33, having lower range as 12 and upper range as 35

1. 13
2. **12**
3. 19
4. 35

upGrad

Poll 9 (30 Sec.)

What is the absolute value of median in the given sorted array:
1 2 3 3 5 5 5 6 7 8 9 11 12 13 15 20

1. 6
2. 8
3. 7
4. 5

Poll 9 (Answer)

What is the absolute value of median in the given sorted array:
1 2 3 3 5 5 5 6 7 8 9 11 12 13 15 20

1. **6**
2. 8
3. 7
4. 5

Poll 10 (15 Sec.)

Given array is: 13 15 16 16 16 17 18 19 20 21 21 23 23 25 26 27 29 30
What is the frequency of occurrence of element 25 in the array?

1. 2
2. 3
3. 1
4. 23

Poll 10 (Answer)

Given array is: 13 15 16 16 16 17 18 19 20 21 21 23 23 25 26 27 29 30
What is the frequency of occurrence of element 25 in the array?

1. 2
2. 3
3. **1**
4. 23

Hands-on Coding

- Finding the smallest missing element from the sorted array of n distinct elements. Elements are in the range of 0 to m-1, where m>n.

- Find median of two sorted arrays.

Homework

1. Count frequency of number in sorted array using binary search.

Tasks to complete after the session

Homework
MCQs
Coding Questions

In the next class...

- Introduction to sorting algorithms
  - Bubble sort
  - Selection sort
  - Insertion sort

upGrad

#RahoAmbitious

Thank You!