



# Tecnológico de Monterrey

---

Competency Evidence

---

## **Programming of data structures**

Inés Alejandro A00834571

Jesus Fong Ruiz A01254062

Profesores: Luis Ricardo Peña Llamas  
Jorge Gonzalez

3/09/2022

Ines Alejandro

We can use the art of programming in our daily lives. As students of computer engineering we learn new steps and methods of search and sort information every day. It's important to learn new methods of organization to analyze algorithms and find the most efficient one that can process the data in a faster way. There are a great variety of sorting and searching algorithms; the bubble sort, sequential search and sorting, etc. Each method has a certain code that makes it different from the others. Some contain algorithms that make the complexity in a more efficient way.

The two main types of searching algorithms are the sequential search and the interval search. The sequential search can be described as an algorithm that traverses sequentially through every element on the vector until it finds the given number. On the other hand, the interval search is a method that targets the center of the vector or list to find the input number. There are other methods that use a different algorithm to sort the data and find the given number within the list. For example, the jumping search is an algorithm that allows the code to check the list for the given number by jumping variables through the list, this means that in the code of the jumping search method we input the number of spaces we want to jump in order to find the given whole number on the list. The negative side of the method is that it is less efficient than using one of the two main methods explained before since the jumping method is programmed for smaller lists and won't be the same efficiency with a long list. But it is similar to the binary search which is one of the two main ones, but with this we can conclude that the binary search is not the most efficient of the two main methods of searching for data in a short list. The most efficient method for searching whole numbers and finding the input would be the linear search which is a method that goes one-by-one in order to find the number in a short list but not in a bigger list.

Jesus Fong

The language of programming is one of the most innovative forms of writing that lets us do tasks like sorting lists or finding a specific entry in an efficient way. With C++ there are hundreds of methods that allow us to arrange and find specific items in the way we write the code. With this we can infer that there are some codes that are more efficient with the information provided than others.

In C++ there are 2 main categories of sorting and searching methods that allow us to perform the tasks in an easy way. For example the groups of the methods are the sequential search and the interval search. These 2 methods of sorting are similar in the idea but different in the way they execute the given tasks. Some methods start analyzing the data from the beginning of the list while other methods may even start from the middle and separate all the data. To start off with the Sequential search it is considered the easiest method to apply as a search engine of a list. What this method does is analyze the list and its numbers one-by-one from the beginning, and then it jumps each number until it finds the desired one. The other method that works as a search engine would be the interval search, and this method is considered a similar approach to the sequential/linear search but a little more complex. What the binary search or interval method does is it parts the list in half in the beginning then it analyzes if the input we gave the code is greater than or less than, then the code works from halfway to find the desired input. Although the linear search is one of the simplest codes to find the desired input, the interval search is considered much more efficient to find the data. With this conclusion we can analyze that the interval code is more efficient because it parts the whole information in half and then analyzes in which side it is and with this we can clear the half of the list without having to go one-by-one compared to the sequential search.

## References

GeeksforGeeks. (n.d.). *Searching Algorithms*. Retrieved September 5, 2022, from

<https://www.geeksforgeeks.org/searching-algorithms/>

GeeksforGeeks. (2022, August 24). *Linear Search Algorithm*. Retrieved September 5, 2022,

from <https://www.geeksforgeeks.org/linear-search/>

GeeksforGeeks. (2022b, September 5). *Binary Search*. Retrieved September 5, 2022, from

<https://www.geeksforgeeks.org/binary-search/>

*Linked list data structure*. (s/f). Programiz.com. Recuperado el 6 de septiembre de

2022, de <https://www.programiz.com/dsa/linked-list>