

Cristian Pintor
2/28/2021

In Python, searching is “the algorithmic process of finding a particular item in a collection of items” (Miller et al., 2006). A search will typically return true or false when searching a list to see if it is present. Meanwhile sorting is “the process of placing elements from a collection in some kind of order” (Miller et al., 2006).

A searching example of when sorting and searching are necessary could be when developing a phone book directory application where you want to sort contacts by last name, first name or age. This type of process could be useful in order to find the contact you are looking for in an efficient amount of time. Sorting and searching could also be used when in a library database for books, where books can be searched and the algorithm can determine whether the book is in stock or not, the location of the book in the library and how many copies are available.

According to Marcus Sanatan, bubble sort is an algorithm that “iterates over a list, comparing elements in pairs and swapping them until the larger elements “bubble up” to the end of the list, and the smaller elements stay at the “bottom””. Another sorting algorithm is selection sort, which segments a list into two parts: sorted and unsorted, which “continuously remove the smallest element of the unsorted segment of the list and append it to the sorted segment” (Sanatan). A third sorting algorithm is insertion sort. This algorithm works by segmenting a list into sorted and unsorted parts and “iterates over the unsorted segment, and inserts the element being viewed into the correct position of the sorted list” (Sanatan).

Two types of searching algorithms are linear search and binary search. Linear search is a simple algorithm which searches an element or value in an array by “traversing the array from the starting, till the desired element or value is found” (“Introduction to Searching Algorithms”). As for in a binary search algorithm, it is used in a sorted array or list and begins by comparing the element to be searched with an element in the middle of the array or list. When there is a match, it returns the index of the middle element, if it does not match then it checks whether the element to be search is less or greater than in value than the middle element (“Introduction to Searching Algorithms”).

References

Introduction to searching algorithms. (n.d.). Retrieved February 28, 2021, from <https://www.studytonight.com/data-structures/search-algorithms>

Miller, B., Ranum, D., & College, L. (2006). 6.6. SORTING. Retrieved February 28, 2021, from <https://runestone.academy/runestone/books/published/pythonds/SortSearch/sorting.html>

Sanatan, M. (n.d.). Sorting algorithms in python. Retrieved February 28, 2021, from <https://stackabuse.com/sorting-algorithms-in-python/#bubblesort>