Eric Li
A20419312
ITMD 413 Shamsuddin
HW 6 - Sorting and Searching Paper

In data and databases, sorting is where data is organized in a specific way. Searching involves looking up the data in the database (Zaveri, 2019). Sorting is when values are ordered (Sorting, searching and algorithm analysis). One example of when sorting and searching are needed in software applications is when someone is logging into their account. The application needs to search for the account in the database and any changes made need to be sorted or organized. Another case could be in a mapping application, where location data needs to be regularly added and updated to reflect any real-world alterations. Searching and sorting are important because they are main proponents of many software applications. Without search algorithms, we would need to go one-by-one to check each value (Why do we need searching algorithms?). Without sorting, searching would also be more difficult.

One type of sorting algorithm is the Selection Sort. This algorithm takes the smallest number in the list and places it in the first spot of the list. Then it takes the second smallest number in the list and places it in the second spot of the list. This process then repeats for the remainder (Deitel, Deitel, 2021). Another type of sorting algorithm is the Insertion Sort. Like Selection Sort, this one is also inefficient. This sorting algorithm starts with the second element in the array and replaces it with the first one if it is less than it. Then, the third will be inserted so that the first three elements are ascending in value (Deitel, Deitel, 2021). A third type of sorting algorithm is Merge Sorting. In Merge Sorting, two sub arrays are created and sorted before being merged back together (Deitel, Deitel, 2021).

One type of searching algorithm is the Linear Search Algorithm. This algorithm goes through the list and checks if the value in the list matches the search key. If it exists, the location of the element will be returned. If it does not exist, a sentinel value is returned (Deitel, Deitel, 2021). A second type of searching algorithm is the Binary Search Algorithm. This algorithm will check the middle element of an array against the key. If the element is greater, then the first half of the array will be taken, and the middle value of that sub array will be checked against the key. If it is less than the key, the second half from the middle element will be checked against the key (Deitel, Deitel, 2021).

Selection and Insertion sort are both very inefficient because the work sequentially through a list. The Merge sort is more efficient since it breaks up an array and checks the values before adding them back to an array (Deitel, Deitel, 2021). The Binary Search is the most efficient search because it quickly eliminates values that could not be the correct value, rather than going one by one and checking each element (Deitel, Deitel, 2021).

References

Deitel, P., & Deitel, H. (2021). *INTRO TO PYTHON FOR COMPUTER SCIENCE AND DATA SCIENCE: Learning to program with ai, big data... and the cloud, global edition*. S.l.: PEARSON EDUCATION LIMITED.

Sorting, searching and algorithm analysis. (n.d.). Retrieved from https://python-textbok.readthedocs.io/en/1.0/Sorting_and_Searching_Algorithms.html

Why do we need searching algorithms? - Searching - KS3 Computer Science Revision - BBC Bitesize. (n.d.). Retrieved from https://www.bbc.co.uk/bitesize/guides/zgr2mp3/revision/1

Zaveri, M. (2019, May 12). An intro to Algorithms: Searching and Sorting algorithms. Retrieved from https://codeburst.io/algorithms-i-searching-and-sorting-algorithms-56497dbaef20