Solano, Ralph Christian 2023 / 3 / 21
CS 201A CS22S2

1. What are the advantages and disadvantages of merge and quick sort.
2. How to compute the time of merge and quick sort.

Two of the most widely employed sorting algorithms are merge sort and quicksort. Both techniques have benefits and drawbacks, and whatever algorithm is used will depend on the particular specifications of the current task.

Merge sort algorithm

The process of sorting an array using the divide-and-conquer approach is called Merge sort. This algorithm repeatedly divides the array into two halves until it cannot be divided further, and then it merges the sorted halves. Merge sort has certain benefits, such as its stable sorting and a worst-case time complexity of O(nlogn). It is also effective for sorting linked lists that have expensive random element access.

However, Merge sort has a couple of drawbacks. Firstly, it needs extra memory to store the subarrays during the merging process, which could be problematic when dealing with significant amounts of data. Secondly, it is not an in-place sorting algorithm, meaning it requires additional memory to store the sorted array.

Quicksort Algorithm

Quicksort is a sorting algorithm that follows the divide-and-conquer approach. It sorts an array by choosing a pivot element and dividing the array around the pivot. The pivot element is positioned so that all elements smaller than the pivot are on its left, and all larger elements are on its right. The pivot element is then sorted recursively with the same process applied to the left and right sub-arrays. The primary advantage of quicksort is its quick average-case time complexity of O(nlogn). However, it has a potential drawback in that its worst-case time complexity can reach O(n^2) if the pivot element is not selected carefully, which can occur when the array is already sorted or contains numerous identical elements. This can be solved by utilizing a randomized version of quicksort that selects a random pivot element. Quicksort is also not stable, which means that the relative order of equal elements may be modified.

In conclusion, both merge sort and quicksort are effective sorting algorithms with distinct advantages and disadvantages. Merge sort is not an in-place sorting algorithm, but it is stable and has a guaranteed worst-case time complexity of O(nlogn). Quicksort is quick and effective for sorting huge datasets, although it is unstable and has a worst-case temporal complexity of O(n2). Many factors, like the size of the dataset, memory restrictions, and the necessity for stability, will determine which of the two techniques is best for the given situation.

References:

https://www.geeksforgeeks.org/quick-sort-vs-merge-sort/?ref=lbp
https://www.geeksforgeeks.org/merge-sort/?ref=lbp
https://www.geeksforgeeks.org/quick-sort/?ref=lbp
https://www.digitalocean.com/community/tutorials/merge-sort-algorithm-java-c-python