Diego Craveiro Chaves

North Metropolitan TAFE  30 Aberdeen St., Perth WA 6000

assessment 2: Hashing, sorting and searching

Table of Contents

[Task 1: Discuss the advantages and disadvantages of 3 sorting algorithms. 2](#_Toc54200216)

[Merge sort 2](#_Toc54200217)

[Quick sort 3](#_Toc54200218)

[Insertion sort 4](#_Toc54200219)

[References 5](#_Toc54200220)

# Task 1: Discuss the advantages and disadvantages of 3 sorting algorithms.

## Merge sort

This algorithm has three outstanding advantages:

* It has the same time complexity for best, worst and average scenarios (), which makes it more predictable than most other sorting algorithms
* Also because of the previous point, merge sort works better with large data sets than algorithms that have slower worst case time complexity, such as quick sort
* It is a good choice to sort data structures that are usually accessed sequentially, such as linked lists

And comes with these disadvantages:

* The way that it is commonly implemented is by creating extra arrays or lists split the original one into smaller chunks, so it is not quite efficient in terms of space complexity ()
* Since it goes through the whole data structure even if it is already sorted, it is less efficient in terms of time complexity for smaller data sets
* Although more predictable due to having time complexity most scenarios, it is not always as fast as algorithms with better best case time complexities, such as insertion sort ()

## Quick sort

Advantages:

* Just like merge sort, this algorithm is quite efficient () on average
* This algorithm also uses a divide-and-conquer approach, which can be implemented elegantly with recursion using fewer lines of code than some other sorting algorithms
* Because it is usually implemented in-place, its space complexity is more efficient than merge sort ()

Disadvantages:

* Worst case scenario time complexity is , which makes it slower than merge sort in such case
* Not quite intuitive at first, because it uses a random pivot which allows several different versions to be implemented (some implementations even use more than one pivot)
* If the elements of the data set to be sorted are integers, it is slower than radix sort, which has time complexity of

## Insertion sort

Advantages:

* Relatively intuitive and easy to implement and understand, since it basically loops through all elements in sequence and compares each one to its predecessors
* Can be relatively fast compared to quick sort and merge sort in the best case scenario for small data sets ()
* Constant space complexity (), since the sorting is done in-place

Disadvantages:

* It is generally slower than most other sorting algorithms ( on average)
* Because of the previous point, it is practically only useful for educational purposes
* If the array or list is already sorted backwards (e.g. the elements are in descending order and it needs to be sorted in ascending order), it still goes through the whole data set and compares each element with the previous ones, which also makes it less efficient in the worst case scenario ()

# References

*Advantages And disadvantages of sorting*. (2014, February 28). Retrieved from Interesting Information: http://z-sword.blogspot.com/2014/02/advantages-and-disadvantages-of-sorting.html

*Answer: Sorting*. (n.d.). Retrieved from Carnegie Mellon University: http://www.cs.cmu.edu/~cburch/211-fa97/review/q-arrays/a-sorting.html

Arora, A. (2019, December 18). *Sorting Algorithms- Properties/Pros/Cons/Comparisons*. Retrieved from Medium: https://medium.com/@ayusharora009/sorting-algorithms-properties-pros-cons-comparisons-1b43599d587b

Bisht, A. (2019, August 23). *Quick Sort vs Merge Sort*. Retrieved from GeeksForGeeks: https://www.geeksforgeeks.org/quick-sort-vs-merge-sort/

*Insertion Sort*. (2020, July 25). Retrieved from GeeksForGeeks: https://www.geeksforgeeks.org/insertion-sort/

*Know Thy Complexities!* (n.d.). Retrieved from Big-O Cheat Sheet: https://www.bigocheatsheet.com/