DATA STRUCTURE AND ALGORITHAM

RIHAM AHAMED ABDUL RAHEEM
HND COMPUTING IDM

Contents

List of figure

List of Table

INTRODUCING

This report will consist of constructive comparison and contrast between mainly 3 basic classification algorithms, namely; classification by insertion, bubble and selection with 3 advanced classification algorithms, namely; fusion, rapid sorting and bulk. Each algorithm of the elementary categorization will be respectively compared with said complementary advanced algorithm. Its complexity and efficiency will be evaluated using the Big (O) algorithm.

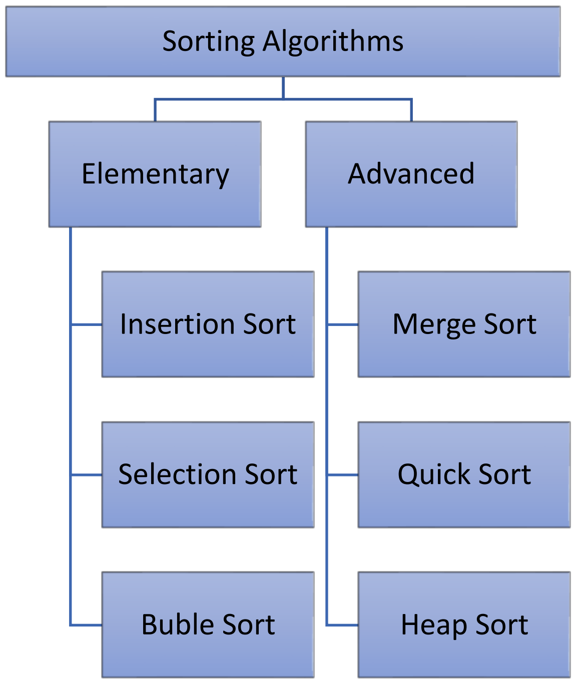

Figure 1: Sorting algorithms

INSERTION SORT VS MERGE SORT

Insertion sort is a simple elementary algorithm that takes the incremental approach as it goes through each element until the sort is complete. As such, this takes quite a bit of time, so insert sorting is only a recommended approach if your dataset is small enough. However, this classification technique can be greatly improved by pre-classifying the data set. This technique is called a binary insert type, where the number of comparisons in the normal insert type is reduced by performing a simple binary search.

Merge sort is an advanced algorithm commonly categorized as a "divide and conquer" algorithm. This is because the technique uses the procedure of dividing the dataset in half, then sorting each half, and finally merging the 2 halves.

If we compare the 2 algorithms, according to the big (o) factors mentioned, we can say that the merge sort works significantly better if O (n * log (n))> O (n2). However, in terms of complexity during implementation, insertion sorting is desirable, since merge sorting involves partitioning the code and performing a merge at the end. However, binary insertion sort works using O (n) linked lists, which is faster, but this technique also involves the implementation of binary search. (Lithmee, 2017)

Table 1: Insertion Sort VS Merge Sort time complexity

	Time Complexity		
	Worst	**Average**	**Best**
Insertion Sort	O(n^2)	O(n)	O(n^2)
Merge Sort	O(n*log(n))	O(n*log(n))	O(n*log(n))

SELECTION SORT VS QUICK SORT

Sorting by selection is a basic sorting algorithm that separates the dataset into 2 data subsets, while one is ordered and the rest is unordered, the algorithm repeatedly finds the minimum value and places it at the beginning of the array neat.

Quick classification is another divide and conquers algorithm, just like merge, it also classifies the dataset before taking any action on it. The first item in the dataset is chosen, then it is placed in the correct position in the dataset, then all items less than that value are placed in front of it while the largest items come after. The general opinion is that quicksort is an unstable algorithm, so it is often recommended to modify it.

As with the previous equation, selection sort is an easier algorithm to implement, but not as efficient as quicksort; however, a basic quicksort implementation is considered unstable. For the algorithm to be stable, consider the indices as a comparison parameter. This adds to the inconvenience during implementation.

That being said, if the dataset involved is a linked list, it is strongly recommended to steer clear of a quick sort and do a combined sort, as linked lists are sequential and merged rankings are also sequentially, while quicksort requires random access. , which is better suited if the dataset is an array. (tutorialstpoint, 2021), (Tutorialstpoint, 2021)

Table 2: Selection Sort VS Quick Sort time complexity

	Time Complexity		
	Worst	**Average**	**Best**
Selection Sort	O(n^2)	O(n^2)	O(n^2)
Quick Sort	O(n^2)	O(n*log(n))	O(n*log(n))

BUBBLE SORT VS HEAP SORT

Bubble sorting is the simplest sorting algorithm available, the procedure is simple; swap the items until the entire data set is in the correct order. This algorithm is an introduction to classification for new programmers. However, it is widely used in the gaming industry because the algorithm can detect subtle changes, such as the missing item, with as little complexity as possible.

Heapsort is an advanced sort algorithm that revolves around the data structure of the binary heap, uses a similar technique to sort by selection, finds the maximum item and places it last and repeats for the rest of the items. Heapsort is another unstable algorithm, to solve this problem, the above mentioned method is used to consider the index as a comparison parameter.

Bubble sort is definitely not designed for particularly large data sets, as it loops through the data set multiple times in a row for each exchange, which is time consuming. Although bubble sorting is an algorithm that is flexible, in the games industry the algorithm is often modified using a queue data structure to increase efficiency.

Heap sort cannot be compared well to bubble sort, because the sheer complexity of implementing a heap sort is enough to conclude that bubble sort should not be compared to it. However, in terms of usability, heap sorting is effective for extremely large data sets, while jitter is problematic, there are many solutions.

Unlike previous advanced algorithms, stack sorting does not partition. (geeksforgeeks, 2021), (Woltmann, 2021)

Table 3: Bubble Sort VS Heap Sort time complexity

	Time Complexity		
	Worst	**Average**	**Best**
Bubble Sort	O(n^2)	O(n^2)	O(n)
Heap Sort	O(n*log(n))	O(n*log(n))	O(n*log(n))

CONCLUSION

Table 4: conclusion of Sorting algorithms

Time	**Size of array**			
	1,000	**5,000**	**10,000**	**50,000**
Insertion	0.002	0.400	1.023	5.928
Selection	0.005	0.560	2.052	7.833
Bubble	0.100	0.890	5.984	12.629
Merge	0.003	0.039	0.972	3.991
Quick	0.034	0.590	1.005	8.385
Heap	0.015	0.088	2.592	7.221

References

geeksforgeeks, 2021. *Comparison among Bubble Sort, Selection Sort and Insertion Sort.* [Online]
Available at: https://www.geeksforgeeks.org/comparison-among-bubble-sort-selection-sort-and-insertion-sort/
[Accessed 17 July 2021].

Lithmee, 2017. *What is the Difference Between Bubble Sort and Insertion Sort.* [Online]
Available at: https://pediaa.com/what-is-the-difference-between-bubble-sort-and-insertion-sort/
[Accessed 17 July 2021].

Tutorialstpoint, 2021. *Data Structure and Algorithms - Quick Sort.* [Online]
Available at: https://www.tutorialspoint.com/data_structures_algorithms/quick_sort_algorithm.htm
[Accessed 17 July 2021].

tutorialstpoint, 2021. *Data Structure and Algorithms Selection Sort.* [Online]
Available at: https://www.tutorialspoint.com/data_structures_algorithms/selection_sort_algorithm.htm
[Accessed 17 July 2021].

Woltmann, S., 2021. *Heapsort – Algorithm, Source Code, Time Complexity.* [Online]
Available at: https://www.happycoders.eu/algorithms/heapsort/
[Accessed 17 July 2021].