FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT
Tınaztepe Yerleşkesi, Buca-Kaynaklar, Dokuz Eylül Üniversitesi, İZMİR, TÜRKİYE
07.04.2023
Assignment: CME 2204 | Assignment 1 | Comparing Merge Sort and Quick Sort
Prepared by: Yusuf Gassaloğlu

1. Introduction

Main scope of the project is to compare Merge Sort and Quick Sort, to see which sorting method is more efficient in which situations.

2. Problems Encountered

Quick sort code thrown stack overflow error because of the Java. Java does not have tail call optimization. Tail recursion is defined as a recursive function in which the recursive call is the last statement that is executed by the function. So basically, nothing is left to execute after the recursion call. So tail recursion method is added manually. The idea of tail recursion is. first solves sub-problem with smaller size, call recursion only when sub-problem is small enough.

3. Overall Assessment of The Assignment

Merge sort is more efficient if the elements in the array are equal. Since the array values are equal, the entire array must be scanned each time with quicksort. Quick sort is more efficient if the elements in the array are random. Since the array values are random, quick sort faster than merge sort. If the array elements are increasing or increasing, less efficient way to sort is quick sort first index is pivot. Because the entire array must be scanned each time.

3.1 Time Complexity

3.1.1 Merge Sort

Merge sort performs the same number of comparison and assignment operations for an array of a particular size. Therefore, its worst-case time complexity is the same as best-case and average-case time complexity, those are:

2-way merge sort: $\mathbf{O(nlog_2 \, \boldsymbol{n})}$
3-way merge sort: $\mathbf{O(nlog_3 \, \boldsymbol{n})}$**.**

3.1.2 Quick Sort

In quicksort, the choice of the pivot plays an important role as seen in the table. Let's suppose we always choose the rightmost element of a list to be the pivot and the input array is reverse sorted. The partitions created will be highly unbalanced (of sizes 0 and (n - 1) for a list of size n); that is, the sizes of the partitions differ a lot. This results in the worst-case time complexity of $\mathbf{O(\boldsymbol{n}^2)}$**.**

3.2 Space Requirement

3.2.1 Merge Sort

Merge sort requires the creation of two subarrays in addition to the original array as seen in the table. This is necessary for the recursive calls to work correctly. Consequently, the algorithm must create n elements in memory. Thus, the space complexity is O(n). Merge-sort can be made in place, but all such algorithms have a higher time complexity than O(n log n).

3.2.2 Quick Sort

Quicksort is an in place sorting algorithm. Its memory complexity is O(1).

4. Comparison Table

	EQUAL INTEGERS			RANDOM INTEGERS			INCREASING INTEGERS			DECREASING INTEGERS		
	1000	**10000**	**100000**	**1000**	**10000**	**100000**	**1000**	**10000**	**100000**	**1000**	**10000**	**100000**
Merge Sort Two Parts	0.66 ms	1.03 ms	11.8 ms	0.72 ms	1.52 ms	23.3 ms	0.03 ms	0.27 ms	3.13 ms	0.02 ms	0.29 ms	4.57 ms
Merge Sort Three Parts	0.36 ms	0.71 ms	4.63 ms	0.06 ms	0.08 ms	9.67 ms	0.03 ms	0.38 ms	2.70 ms	0.31 ms	0.53 ms	4.37 ms
Quick Sort First Index Pivot	2.50 ms	25.1 ms	728.2 ms	0.49 ms	1.07 ms	6.05 ms	0.13 ms	12.3 ms	941.9 ms	0.30 ms	27.9 ms	2889 ms
Quick Sort Random Index Pivot	2.24 ms	39.1 ms	1244 ms	0.11 ms	0.83 ms	9.40 ms	0.05 ms	0.56 ms	5.86 ms	0.05 ms	0.57 ms	6.06 ms
Quick Sort Median Index Pivot	0.45 ms	0.60 ms	6.08 ms	0.07 ms	0.49 ms	6.06 ms	0.01 ms	0.11 ms	1.43 ms	0.01 ms	0.20 ms	1.99 ms

5. Questions

5.1

You have a Turkish-English dictionary with a single word for each word in alphabetical order and you want to translate it into an English-Turkish dictionary. For example; if your Tur-Eng dictionary contains [ayı : bear, bardak : glass, elma : apple, kitap : book] then your Eng-Tur dictionary should be [apple : elma, bear : ayı, book : kitap, glass : bardak]. If we think that there are thousands of words in your dictionary, which sorting algorithm do you use to do this translation faster?

Answer: I will use quick sort algorithm median index is pivot. It is the fastest way to sort random distributed arrays. As seen in the table above, the sort algorithm sorted random integers in 6.06 ms. Quick sort algorithm is 3.5 times faster than 2 way merge sort.

5.2

When you inquire Sub-Upper Pedigree, an ordered list of people according to their birth date comes out in the system of e-Devlet. However, you want to rank the people from the youngest to the oldest one. If you are asked to do this operation using a sorting algorithm, which algorithm do you use?

Answer: Array will be sorted in descending order, because younger person’s birth date is bigger than older. Radix sort algorithm is the best for this situation. Radix sorting requires more memory and space. But this report compares merge and quick sort. So, I will use quick sort algorithm median index is pivot. It is the fastest way to sort ordered arrays. As seen in the table above, the sort algorithm sorted the sorted integers in 0.01 ms, 0.11 ms, 1.43 ms, 0.01 ms, 0.20 ms, 1.99 ms. Quick sort algorithm is much faster than merge sort.

6. References

https://www.geeksforgeeks.org/middle-of-three-using-minimum-comparisons/
https://www.geeksforgeeks.org/3-way-merge-sort/
https://www.geeksforgeeks.org/quick-sort/?ref=lbp
https://www.geeksforgeeks.org/quicksort-tail-call-optimization-reducing-worst-case-space-log-n/
http://www.cs.nthu.edu.tw/~wkhon/algo08-tutorials/tutorial2b.pdf
https://www.mycareerwise.com/programming/category/sorting/3-way-merge-sort
https://stackoverflow.com/questions/53354898/tail-call-optimisation-in-java
https://www.geeksforgeeks.org/tail-recursion/
https://www.interviewkickstart.com/learn/quicksort-vs-merge-sort#:~:text=Merge%20sort%20is%20an%20external,memory%20throughout%20the%20sorting%20process.
https://www.algolist.net/Algorithms/Sorting/Quicksort

07.04.2023
CME 2204

Yusuf Gassaloğlu
2020510034