iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** I **Month of publication:** January 2015

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Performance Analysis of Sorting Algorithms with C#

Muhammad Usman[1], MudassarAfzal[2], Zaman Bajwa[3]
Department of Computing & Technology, Iqra University

Abstract --- Sorting technique is mostly used in computer sciences and it used as a scale of a system performance and efficiency through implementing different sorting algorithm's. In this research paper we have focus on the performance of different sorting algorithms which are measured in term of time complexity i.e execution time of sorting algorithm, amount of memory required. To prove the best result among the different sorting algorithms we have use the compilers of different computer programming languages and collect the results of Time complexity to measures the difference among them.
Keywords --- Sorting Algorithms, Operating System, Algorithm Performance, Graphs, Information Technology, Hardware

I. INTRODUCTION

Sorting is the most uttermost algorithmic problem. It's the most methodically studied problem in computer sciences. Sorting algorithm is an algorithm that puts elements of a list in a certain order. The sorting algorithms used to arrange the data in numerical order and lexicographical order. Efficient sorting is important to get maximum throughput of the system and ultimately that system throughput provide maximum efficiency to our end user [4]. It is also frequently useful for canonicals statistics and for generating human coherent output. Since the initiate of computing the sorting concern has demand a immense deal of research work, and researcher have focused to large extent to find and optimize the best sorting algorithm [5]perhaps due to the complications of solving it proficiently despite its simple acquainted statement.
In this paper we implemented various sorting algorithms i.e Quick Sort, Bubble Sort, Insertion Sort on different Compilers i.e C family which include C++ and provide the comparison on the basis of time complexity for the best sorting algorithm among the above because sorting proficiency is appropriate to almost all applications.

II. RELATED WORK

The key point of sorting technique was to compare the runtime performance of a microprocessor on integer sorting, a common general-purpose computing scale.

A. In most practical situations the best sort to use is the one provided by the standard (e.g., Java, C, or C++) libraries. We study sorts because general sorts do not work in all situations and because sorting is a simple illustration of algorithmic techniques. In this paper we have investigated the origins of bubble sort and its enduring popularity despite warnings against its use by many experts. We confirm the warnings by analyzing its complexity both in coding and runtime.

B. In this paper, we present an efficient comparison-based sorting algorithm for many-core GPUs. We compare our results to state-of-the-art algorithms. In practice, our results demonstrate up to 30% better performance than previous optimized comparison-based algorithms for input sequences with millions of elements. As far as we know, ours is the fastest comparison-based sorting algorithm for modern many-core GPUs.

C. In this paper we analysis the performance of quick sort and insertion sort. Insertion sort is suitable for small files, but it is an O(n2) algorithm, but with a small constant. Also note that it works best when the file(/numbers) is already almost sorted. Quick sort amazed me. Didn't you get amazed?! Well it is an O(n*log(n)) algorithm on an average case and an O(n2) algorithm in the worst case scenario.

III. SORTING ALGORITHM

A. Bubble Sort

It is a simple n easy sorting algorithm used in computer science algorithm. It starts with evaluate with the first two elements and if first element is greater than second element then swap it. It continues for each pair of element to the end of data set. It again starts with the first two elements and repeating until no swap has occurred in the last pass. If we have many numbers of elements then the total elements of comparison is very huge. This algorithm is seldom used but in education.

B. Insertion Sort

It is a competent algorithm for sorting a small number of elements. The insertion sort is like its name suggests inserts each element into its suitable place in the final list [3]. Sorting a hand of playing card is one of the real time examples of insertion sort. It sort small array fast but large array it sort vary slowly.

C. Quick Sort

Quick sort is also known as partition exchange sort. Quick sort can sort a list of data elements much faster than any of other sorting algorithms. Quick sort algorithm is based on the actuality that it is faster and easier to sort two small arrays than larger one [5]. Quick sort is based on divide and conquer method. The most express rival of quick sort is heap sort. The heap sort is slower than the quick sort.

Analysis of different Sorting Algorithms

TIME COMPLEXITY				
SORT ALGORITHM NAME	BEST	AVERAGE	WORST	SPACE COMPLEXITY
Bubble Sort	O(n)	----	O(n2)	O(n)
Insertion Sort	O(n)	O(n2)	O(n2)	O(n)
Quick Sort	O(logn)	O(nlogn)	O(n2)	O(n + log n)

IV. EXPERIMENTAL WORK

In experimental work we have performed the LAB experiments by using different Hardware environments and with different Operating systems and we have determined the efficiency of sorting algorithms by depending upon speed and time utilization of performance. The results show that efficiency of Sorting Algorithms change with the use of different Hardware, Operating System and as well the time of execution has significant impact on performance of sorting. The below mentioned graphs and the table show the desire results after implementation.

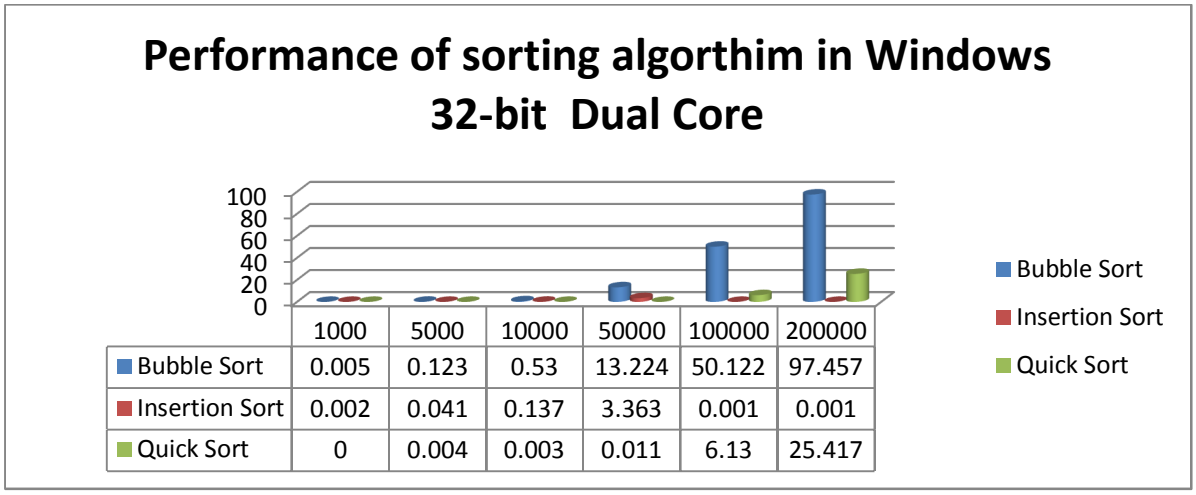
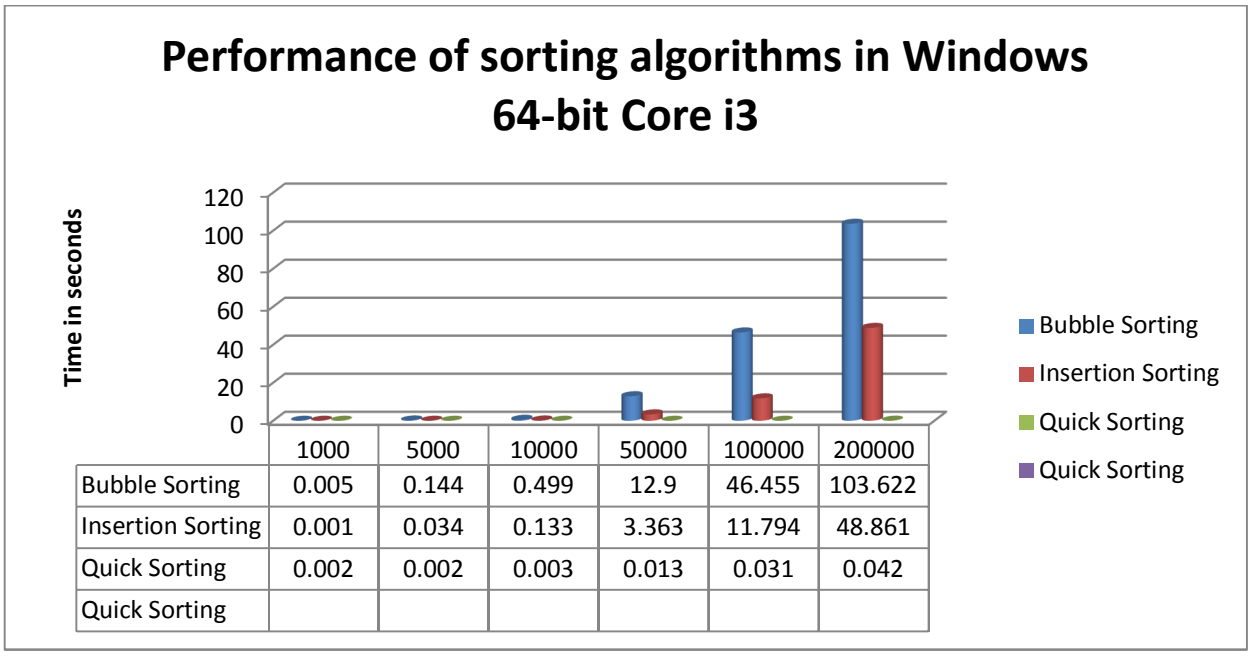


Fig: 1

In Fig: 1, we analyze the performance of different sorting algorithms in windows 32 bit operating system which is dual core. This performance is based on time. In this fig, the insertion sort is the best sorting algorithm due to it takes less time against the bubble sort and quick sort.

Fig: 2

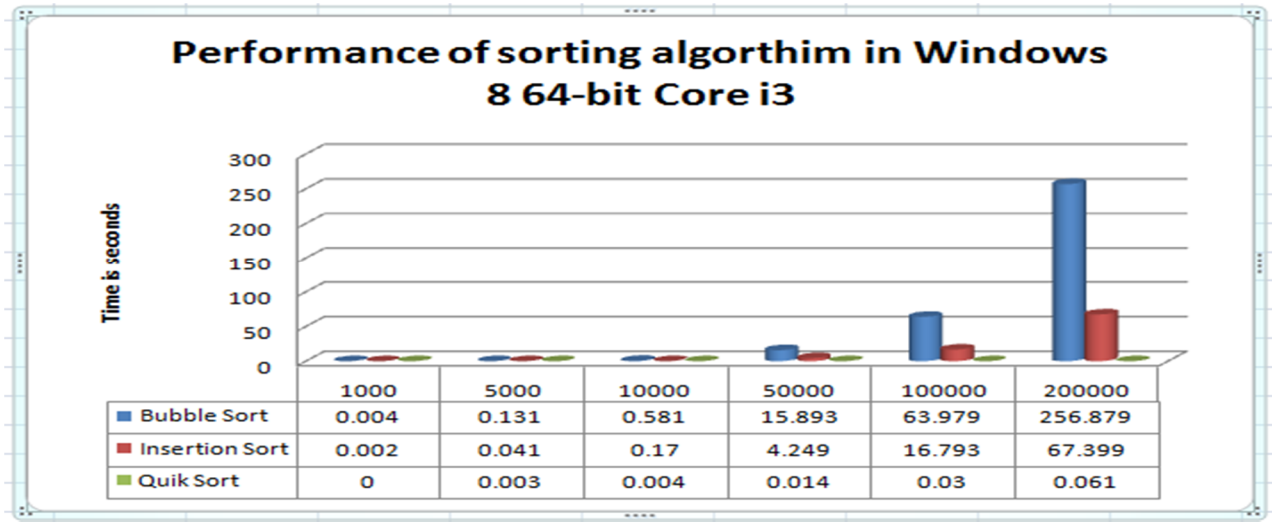
In Fig:2 now, we analyze the performance of different sorting algorithms in windows 32 bit operating system which is core i3. This performance is based on time. In this fig, the quick sort is the best sorting algorithm due to it takes less time against the bubble sort and insertion sort.



	1000	5000	10000	50000	100000	200000
Bubble Sorting	0.005	0.144	0.499	12.9	46.455	103.622
Insertion Sorting	0.001	0.034	0.133	3.363	11.794	48.861
Quick Sorting	0.002	0.002	0.003	0.013	0.031	0.042
Quick Sorting						

Fig: 3

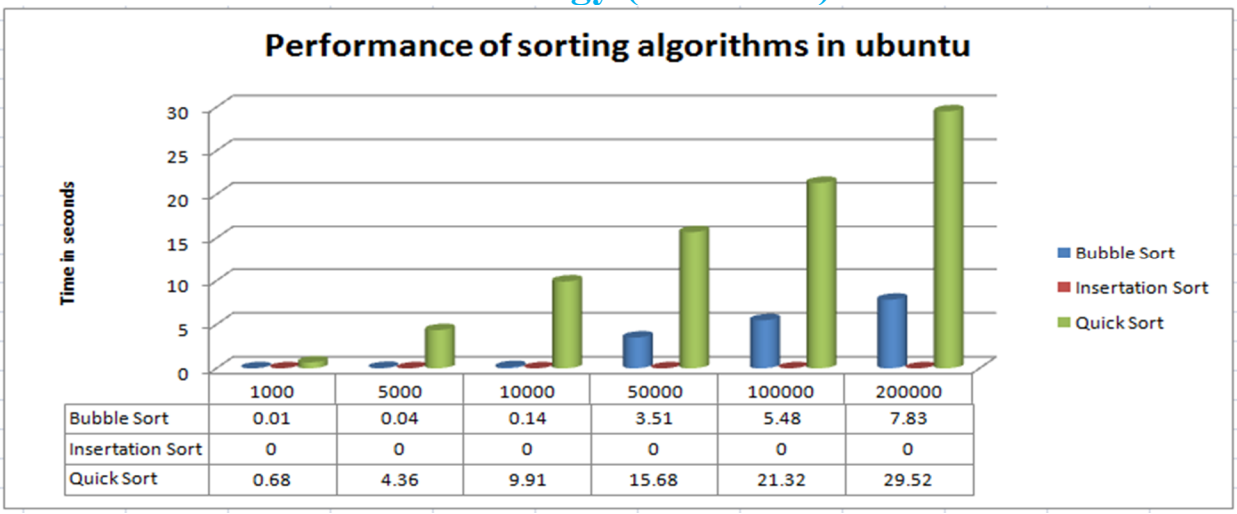
In Fig: 3 now, we analyze the performance of different sorting algorithms in windows 64 bit operating system which is core i3. This performance is based on time. In this fig, the quick sort is the best sorting algorithm due to it takes less time against the bubble sort and insertion sort.



	1000	5000	10000	50000	100000	200000
Bubble Sort	0.004	0.131	0.581	15.893	63.979	256.879
Insertion Sort	0.002	0.041	0.17	4.249	16.793	67.399
Quik Sort	0	0.003	0.004	0.014	0.03	0.061

Fig: 4

In Fig:4 now, we analyze the performance of different sorting algorithms in windows 8 64 bit operating system which is core i3. This performance is based on time. In this fig, the quick sort is the best sorting algorithm due to it takes less time against the bubble sort and insertion sort.



	1000	5000	10000	50000	100000	200000
Bubble Sort	0.01	0.04	0.14	3.51	5.48	7.83
Insertation Sort	0	0	0	0	0	0
Quick Sort	0.68	4.36	9.91	15.68	21.32	29.52

Fig: 5

In Fig:5 now, we analyze the performance of different sorting algorithms in Linux operating system which flavor is ubuntu. This performance is based on time. In this fig, the insertion sort is the best sorting algorithm due to it takes less time against the bubble sort and quick sort.

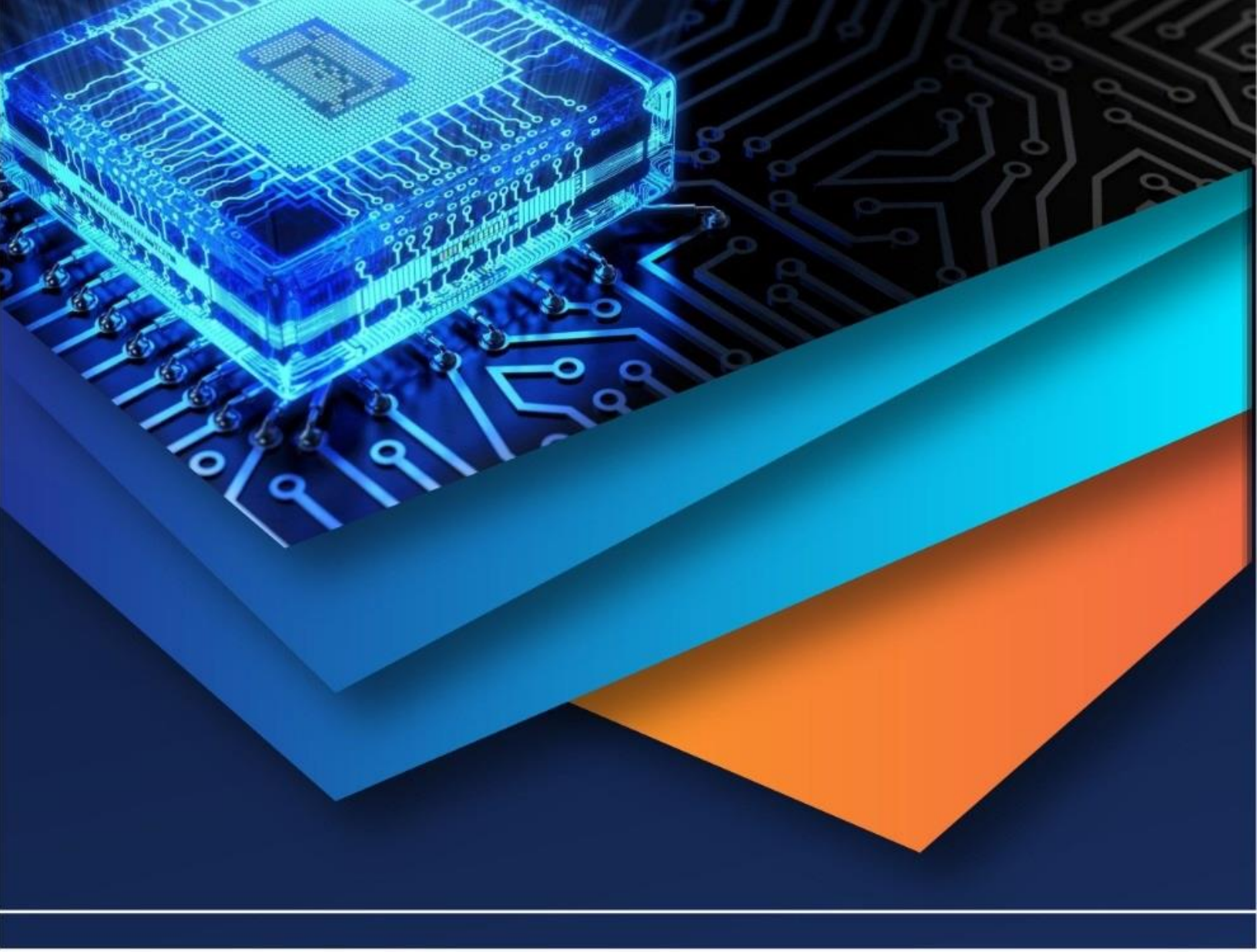
V. CONCLUSION AND FUTURE WORK

In this research paper, we discuss and analyze the performance of different sorting algorithms. The analysis is based on time. In all experiments, there is no best sorting algorithm because in every environment the execution time is change on every execution. In these experiments insertion sorting is the best sorting algorithm in ubuntu environment and worst sorting algorithm is quick sort. In all windows environment the quick sort is best other two sorting algorithms and worst algorithm is bubble sort.

As for future work, we work in other languages like C, C# n Java and other language to check the time and memory utilization.We also plan to work other operating systems.

REFRENCES

[1] Performance Analysis of Sorting Algorithms, Hema .A.M., Associate Professor, Department of Computer Science, Thiagarajar College, Madurai
[2] High Performance Comparison-Based Sorting Algorithm on Many-Core GPUs, Xiaochun Ye, Dongrui Fan, Wei Lin, Nan Yuan, Paolo Ienne
[3] Time Comparison of Quick Sort, Insertion Sort and Bubble Sort.
[4] Bubble Sort: An Archaeological Algorithmic Analysis, Owen Astrachan, Computer Science Department, Duke University
[5] Enhanced Quicksort Algorithm, Rami Mansi, Department of Computer Science, Al al-Bayt University, Jordan,The International Arab Journal of Information Technology, Vol. 7, No. 2, April 2010



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY