Time Analysis Of Sorting Algorithms

Bubble Sort and Quick Sort

By:

PARAS RAJ PAHARI

PROBLEM

In this study I am going to analysis the running time of sorting algorithms. I will be taking any two sorting algorithm, and compare their running time with varied size of arrays. This way I can examine the behavior of sorting algorithms with given different criteria's and see out the quicker and efficient one among these algorithm. Two sorting algorithms that will be compared are **Bubble Sort algorithm** and **Quick Sort algorithm**.

BACKGROUND RESEARCH

Quick sort is Divide and conquer algorithm. Working of quicksort is done by selecting the pivot first, and then quicksorting the elements on the left and quicksorting the elements on the right. It's generally fast with large number of inputs but if it is fed with already sorted inputs it gets slower than usual.

Best Case of Quick Sort is O (nlogn).Average Case of Quick Sort is O (nlogn).Worst Case of Quick Sort is O (n^2).Running times of quicksort also depends on the how the pivot is selected.

Bubble Sort, which also has another name "Sinking Algorithm". Depending on how it's written, Bubble sort generally bubbles the lowest value to the top or sinks the highest value to the bottom. The Bubble sort makes multiple passes through the

list. It is very inefficient for large collection of items. Only way the bubble sort works quicker is if the items are already in sorted order.

Best Case of bubble sort is O (N). That's when array are already in sorted order. Its Average case and Worst case is O (n^2).

HYPOTHESIS

Looking into the background research of quick sort and bubble sort, it's not hard to predict that quick sort comes out on top and is more practical in contrast to bubble sort. Comparing the average case of both, Quick sort have O (nlogn) and Bubble Sort have O (n^2). Since both worst case and average case of bubble sort are same O(n^2), bubble sort is predicted to be impractical with the increase in size of large number of items. Although, bubble sort might be good enough sorting the sorted order of items, it still needs to pass through each items. Quick sort might have worse performance sorting the already sorted list.

EXPERIMENT

Experiment will proceed the implementation of sorting algorithms to compare their running time with different array size, and analyze the result.

Language to be used: JAVA

Software Development Kit: ECLIPSE

Step to be taken for the implementation of this project of comparing running times of sorting algorithms:

i) Generate Random Array with different sizes. Array Sizes are 20000; 40000; 60000; 80000; 100000.

ii) These Randomly Generated arrays of different sizes are fed to bubblesort() and quicksort() to sort them and record the running times in milliseconds.

iii) Also, after sorting those randomly generated arrays, we will fed those sorted items of different array sizes again to bubblesort () and quicksort() algorithms to analyze their behavior with sorted items by recording their running time.

RESULTS

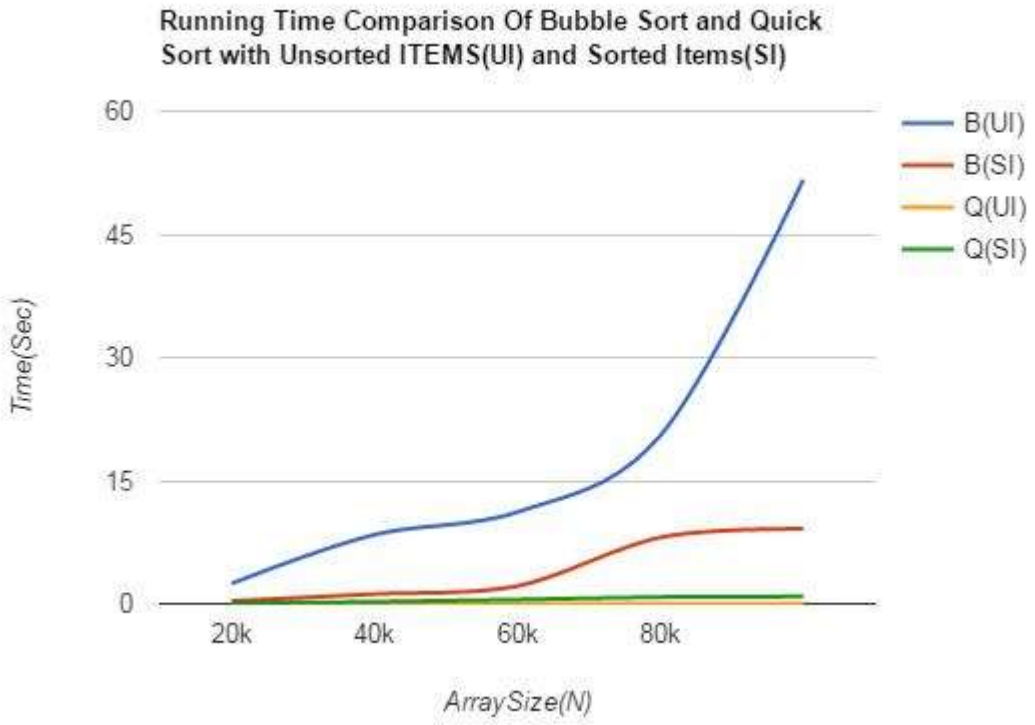
After proceeding through the implementation project of comparing the running time of bubble sort and quicksort algorithm, results are recorded. The record of outcome is shown in the tabular form and the graph below.

Results for Running Time of Bubble Sort with Unsorted Items and Sorted Items

RUNNING TIME OF BUBBLE SORT				
	UNSORTED ITEMS		SORTED ITEMS	
Array SIZE (N)	Time (ms)	Time(sec)	Time(ms)	Time(sec)
20,000	2467	2.467	344	0.344
40,000	8420	8.420	1185	1.185
60,000	1164	1.164	2155	2.155
80,000	20484	2.0484	8086	8.086
100,000	51655	5.1655	9115	9.115

Results for Running Time of Quick Sort with Unsorted Items and Sorted Items

RUNNING TIME OF QUICK SORT				
	UNSORTED ITEMS		SORTED ITEMS	
Array SIZE (N)	Time (ms)	Time(sec)	Time(ms)	Time(sec)
20,000	7	0.007	98	0.098
40,000	6	0.006	246	0.246
60,000	9	0.009	511	0.511
80,000	11	0.011	821	0.821
100,000	14	0.014	889	0.889



CONCLUSION

From the results above, it's amusing how good quicksort with large number of items. With this implementation result it can be seen that hypothesis was right with the prediction that quicksort will come on top. Bubble Sort is really poor and very inefficient when it comes with large size of items to be sorted. Even though bubble sort did pretty well with sorted items. In my opinion, we can probably implement bubble sort to test whether the items are sorted correctly or not. Also, we can see quick sort running time increased with the input of sorted items proving that quick sort doesn't work pretty well with sorted items even though it still did better job than Bubble Sort with Sorted Items.

With this, we can come to the conclusion that quick sort is much faster and quicker than bubble sort with increasing number of size of items. So it's very impractical to use bubble sort for large number of items instead of quicksort. Why choose bubble sort if quick sort can sort large items in a snap. I would choose quick sort over bubble sort any day.