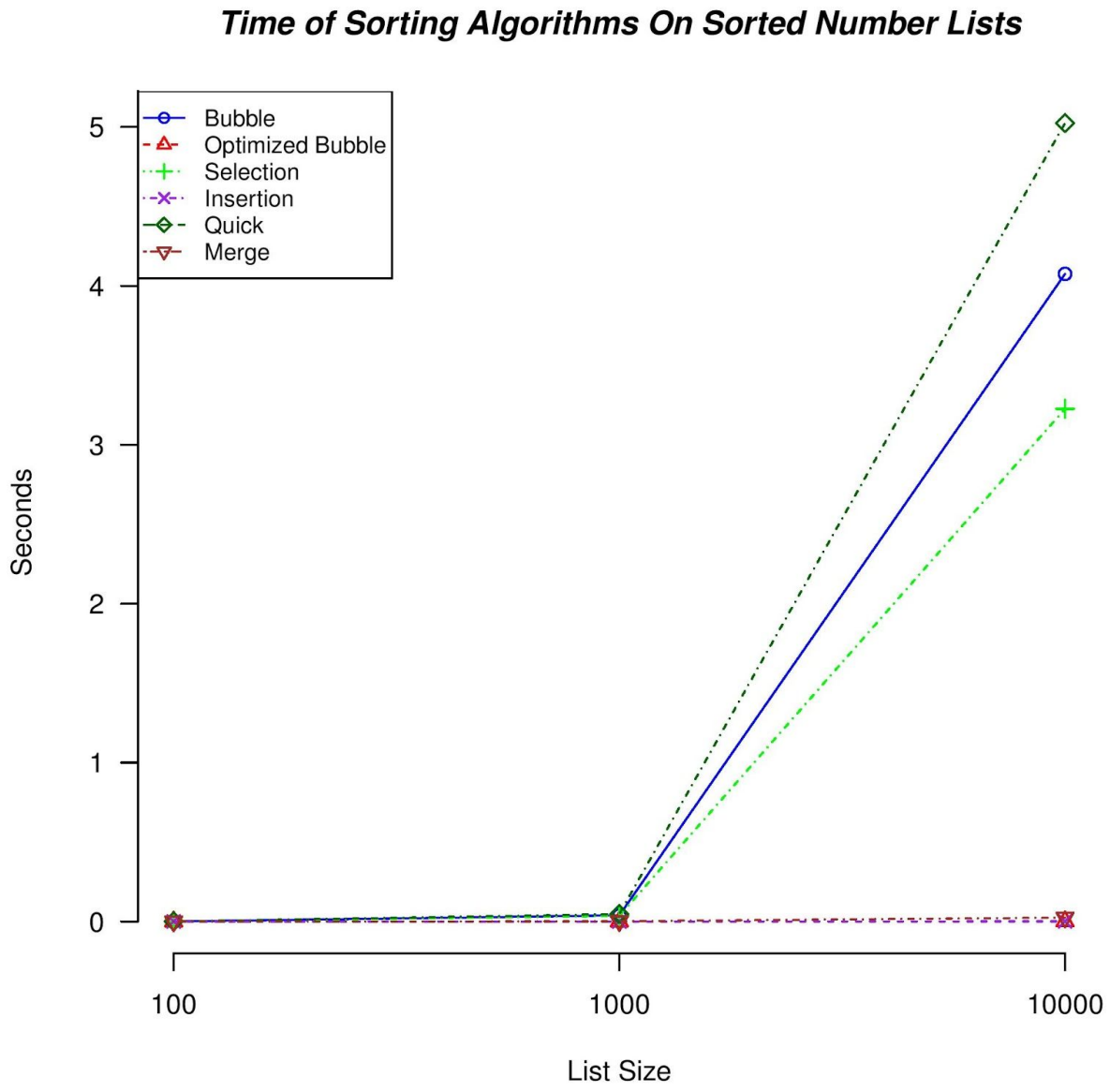
Ervin Hennrich
CMP SCI 3130

Abstract

Six sorting algorithms were tested: Insertion Sort, Bubble Sort, Optimized Bubble Sort, Selection Sort, Merge Sort and Quicksort, to see whether experimental results matched the theoretical. In all cases, the theory did match the experimental results. Although for smaller number sets the time is less pronounced between them. Even for less time efficient algorithms, such as Bubble Sort, it finished within .07 seconds for lists less than 1000. This is expected, but demonstrates for smaller sized sets there is little practical difference between them unless absolute efficiency is necessary. For larger sets, Merge Sort is consistently among the fastest making it one of the best general algorithms for sorting, that was tested. Although, for already sorted lists Optimized Bubble Sort and Insertion Sort beat Merge Sort. For smaller lists, Insertion sort was consistently among the fastest and is the best in general for small lists.

Algorithms on Sorted Lists

Algos w/ sorted lists	100 elements	1000 elements	10000 elements
Selection	0.000332117080688 47656 seconds	0.035737752914428 71 seconds	3.225963115692138 7 seconds
Insertion	1.668930053710937 5e-05 seconds	0.000173330307006 83594 seconds	0.001547336578369 1406 seconds
Bubble	0.000372171401977 53906 seconds	0.038491725921630 86 seconds	4.075166463851929s econds
Optimized Bubble	9.059906005859375 e-06 seconds	0.000111818313598 63281 seconds	0.000853300094604 4922 seconds
Quick	0.000459432601928 71094 seconds	0.047556877136230 47 seconds	5.024553775787353 5 seconds
Merge	0.000171422958374 02344 seconds	0.001959085464477 539 seconds	0.025116920471191 406 seconds



For all of the algorithms tested, the experimental results matched what was expected from the theoretical. It was expected that Optimized Bubble Sort would be one of the fastest for sorting already sorted lists, since it specifically tests for sorted lists. An already sorted list is the best case for this algorithm with an efficiency of Ω(n) which matches the results. It was significantly faster than any other algorithm tested for all list sizes.

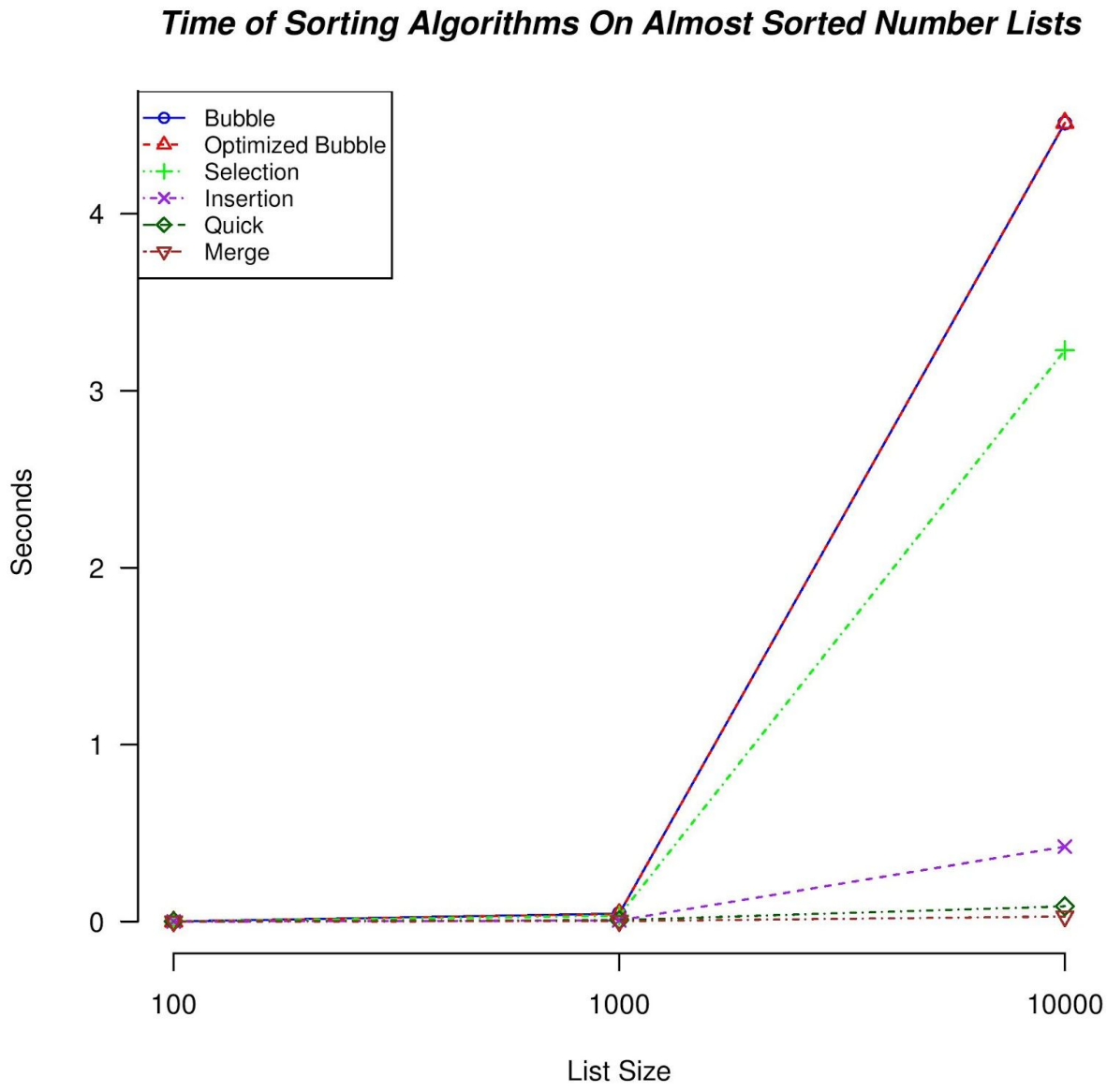
Insertion Sort is similar to Optimized Bubble Sort, its best case is sorted lists and complexity is Ω(n). It also performed extremely well, finishing before Mergesort.

Quicksort is also an interesting case for sorted lists, which are it's worst case (O(n^2)). It was expected to be inefficient but, it was not expected to be the slowest through every run, as

selection sort has a similar efficiency (O(n^2)). This makes quicksort the worst to use for checking for sorted lists.

Algorithms on Almost Sorted Lists

Algos w/ sorted lists	100 elements	1000 elements	10000 elements
Selection	0.000329256057739 2578 seconds	0.033197879791259 766 seconds	3.228452205657959 seconds
Insertion	8.463859558105469 e-05 seconds	0.006320476531982 422 seconds	0.423505783081054 7 seconds
Bubble	0.000413417816162 1094 seconds	0.038491725921630 86 seconds	4.513240337371826s econds
Optimized Bubble	0.000132799148559 5703 seconds	0.042923927307128 906 seconds	4.514485597610474 seconds
Quick	0.000397205352783 2031 seconds	0.008488893508911 133 seconds	0.086304664611816 4 seconds
Merge	0.000197887420654 29688 seconds	0.002401828765869 1406 seconds	0.028995752334594 727 seconds

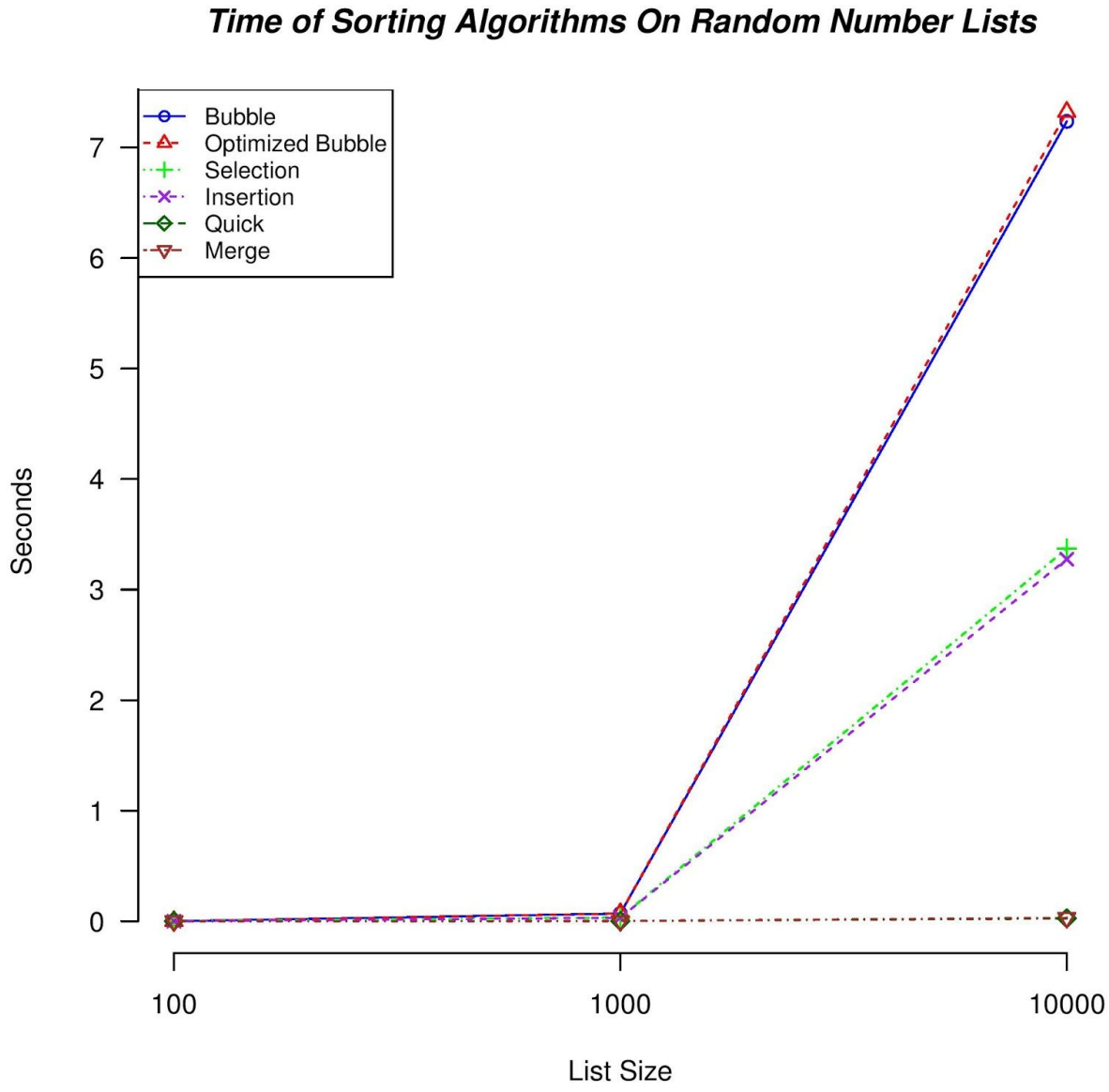


For both almost sorted and random lists Mergesort was the best performing algorithm with Quicksort close behind it. This is expected as both have a similar efficiency (Θ(n log n)) for both nearly sorted and random lists. Both Bubble and Optimized Bubble sorts finished last, with Bubble sort finishing very slightly before, both have a similar complexity of (Θ(n^2)).

For almost sorted lists, Insertion sort preformed much better than on random lists. With an average efficiency of n^2 (Θ(n^2)) which matches the experimental results. Insertion sort finished sorting the 100 element list far before any others, but started to fall behind Quicksort and Mergesort at the 1000 element lists. Insertion sort is consistently one of the fastest for small lists, but is beat out by algorithms with slower growth (like Quicksort and Mergesort ((Θ(n log n))) for larger lists (<10000).

Algorithms on Random Number Lists

Algos w/ sorted lists	100	1000	10000
Selection	0.000307321548461 91406 seconds	0.033943414688110 35 seconds	3.370923995971679 7 seconds
Insertion	0.000326395034790 03906 seconds	0.032087802886962 89 seconds	3.275688409805298 seconds
Bubble	0.000642061233520 5078 seconds	0.069381713867187 5 seconds	7.236100673675537 seconds
Optimized Bubble	0.000649213790893 5547 seconds	0.069781541824340 82 seconds	7.320969820022583 seconds
Quick	0.000132083892822 26562 seconds	0.002079963684082 0312 seconds	0.025364637374877 93 seconds
Merge	0.000195741653442 3828 seconds	0.002307176589965 8203 seconds	0.031631708145141 6 seconds



The results for the random number lists are again exactly as predicted by theory. This should be the average case for each of the algorithms. Bubble, Optimized Bubble, Insertion and Selection sorts all have a Θ(n^2) time complexity. Insertion and Selection sort finish significantly before the Bubble sorts for the 10000 element lists due to them being more efficient, as seen by their finish times for the 100 and 1000 element lists but are still increasing quadratically. The two fastest algorithms Merge sort and Quicksort both have Θ(n log n) time complexity and are the slowest growing functions of the six tested here and therefore finish well before any others.