Performance of Sorting Algorithms

Abhiram R, CS10B060

January 25, 2012

Introduction

This report presents a real-time performance analysis of three of the most famous sorting algorithms, *Bubble Sort*, *Merge Sort*, and *Insertion Sort*, using two different data structures, the *ArrayList* and the *LinkedList*. The Programming language used is JAVA. All testing and analysis has been done on a *Core i5 480M* processor, with 4 GiB RAM, and on the Ubuntu 11.10 Operating System.

Algorithmic Implementations

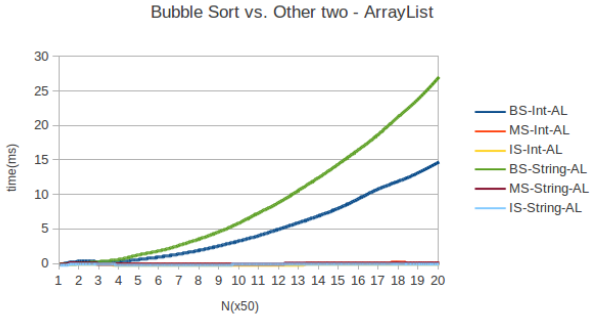
In order to analyze the performance of the algorithms, for data types that are not just integers, the program uses a generic data type that extends the Comparable class. The Standard versions of these algorithms have been implemented. The test procedure for their comparison is as follows. For various values of N, the input size, lists(Array/LinkedList of length N) of random numbers/strings are generated. All the three algorithms are allowed to operate on each list (on each of the ArrayList and LinkedList representations of the same list), and the operation time is measured using the *System.nanotime()* call. Averages are obtained, results are compared and the corresponding graphs are plotted.

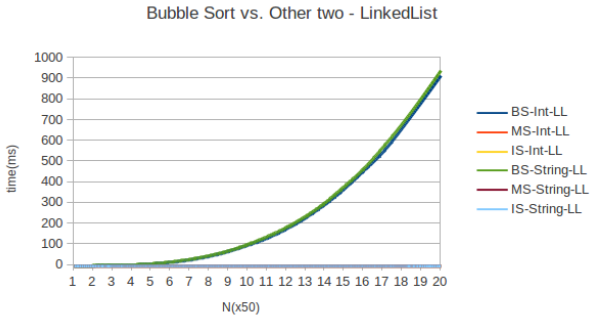
Experimental Results

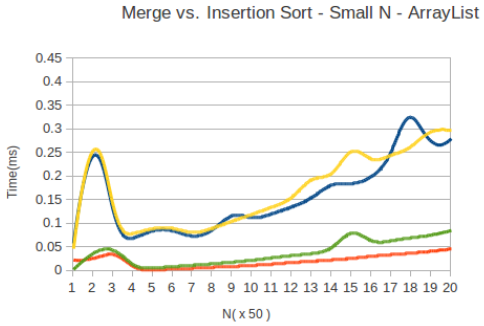
Small Values of N

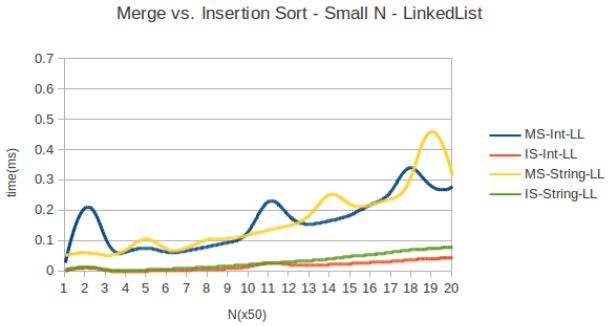
For smaller values of N (upto the order of 1000), Bubble Sort shows a clear performance lag, compared to merge and insertion sorts. A surprising result is that for these smaller values of N, insertion sort shows a better performance than merge sort! Even though we had derived in class that Merge Sort works at O(n lg(n)) and Insertion Sort at $O(n^2)$, the constant terms seems to dominate in MergeSort for smaller values of N. Also, the Linked List data structure seems to have a highly negative impact on the performance of bubble sort, with the ArrayList working a lot better. This makes sense as the swaps that happen in arrays are lot more efficient than those that happen in linked lists. The performance of the other two algorithms is almost unaffected by the nature of the data structure used. Bubble sort seems to handle

integers better than strings with the ArrayList Implementation, whereas with the LL implementation, the performance with both data types is almost similar.









N	BS-Int-AL	MS-Int-AL	IS-Int-AL	BS-String-AL	MS-String-AL	IS-String-AL
50	0.0787752	0.0596816	0.024851733	0.060862433	0.048504133	0.0043982
100	0.717425867	0.370350133	0.021665067	0.1424933	0.391486067	0.044857633
150	0.301863567	0.0680914	0.052671533	0.477556533	0.0740198	0.058137767
200	0.412607667	0.067225067	0.002591667	0.6861576	0.079262133	0.0059452
250	0.785523933	0.090898467	0.0037994	1.5209935	0.093192233	0.007250667
300	1.108375033	0.089377533	0.005140433	1.953587467	0.0942285	0.010567867
350	1.5660831	0.069867967	0.0068219	2.831978967	0.078052533	0.013274467
400	2.0748233	0.083705967	0.009297433	3.719891033	0.091671633	0.016300267
450	2.7445082	0.129683033	0.0104828	4.7764945	0.107316633	0.020027333
500	3.450205533	0.1081116	0.013171133	6.017385967	0.121156467	0.023735533
550	4.2184065	0.123044433	0.015582767	7.6138885	0.136931167	0.029433
600	5.1922267	0.136914467	0.019624133	8.923528867	0.151519833	0.034267667
650	6.095037633	0.153896233	0.0215582	10.8788951	0.206381233	0.037787933
700	7.118775633	0.191339567	0.024895167	12.63408457	0.191866933	0.0436388
750	8.1546399	0.183026267	0.0275638	14.6470028	0.278018367	0.096727
800	9.602820633	0.197242167	0.032966267	16.71147747	0.225203467	0.0571827
850	11.1051251	0.2384954	0.035883267	18.86335357	0.250148433	0.065716533
900	12.0829637	0.3868894	0.039169933	21.64524217	0.2566174	0.072279333
950	13.30820683	0.2394007	0.0425063	23.89473507	0.308386367	0.0762003
1000	14.87597983	0.281481333	0.0480663	27.17662257	0.298236633	0.0871019

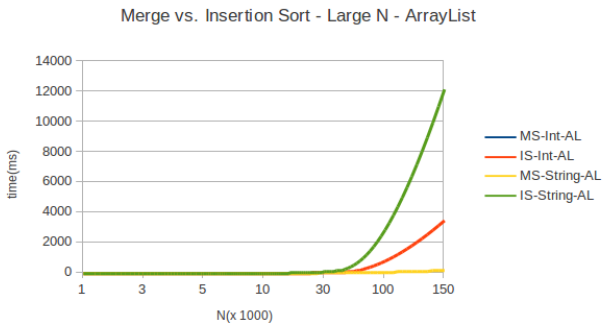
[1]

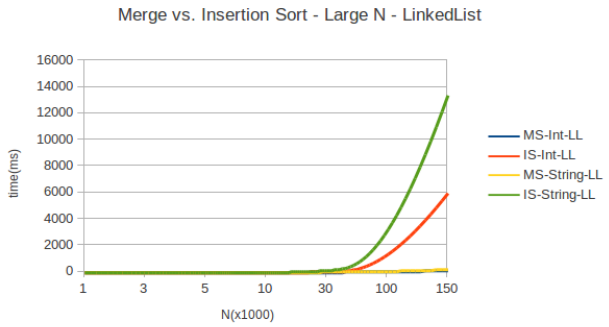
[1]BS - Bubble Sort; MS - MergeSort; IS - InsertionSort; AL - ArrayList; LL - LinkedList; All tabulated values are in milliseconds

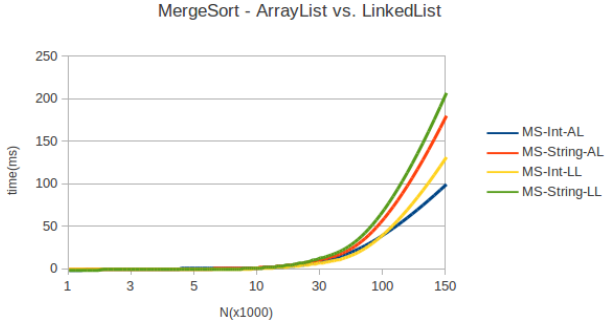
N	BS-Int-LL	MS-Int-LL	IS-Int-LL	BS-String-LL	MS-String-LL	IS-String-LL
50	0.0962899	0.0310563	0.003279367	0.0357464	0.054613233	0.007499533
100	0.329594833	0.332810233	0.022405067	0.400725233	0.0725877	0.024615667
150	1.217477867	0.0436636	0.001615733	1.477151833	0.045015533	0.003172733
200	3.709819433	0.067257767	0.002587133	4.1689706	0.0689879	0.005206833
250	8.495788533	0.087808467	0.0038828	9.281110767	0.130854867	0.007101067
300	15.70377707	0.058684167	0.004962867	17.7702838	0.0608962	0.010001333
350	27.99912883	0.0705033	0.006854067	30.6472022	0.0781911	0.013130667
400	44.8220194	0.083818267	0.0088111	47.75725137	0.114016367	0.016105033
450	68.2816557	0.098596467	0.0111855	70.76371943	0.107659667	0.019654067
500	99.22607077	0.109313	0.0144241	101.603868	0.121964233	0.024488633
550	131.0042766	0.2818772	0.037817033	139.4653905	0.140077667	0.0291422
600	177.0279929	0.167220233	0.021984567	183.732925	0.152677367	0.033403267
650	229.7312587	0.152955167	0.0224586	237.517356	0.172655767	0.038181967
700	295.3488987	0.170102733	0.024852233	300.9223635	0.2888362	0.042368
750	367.8209321	0.183435367	0.027725233	383.0635054	0.206203633	0.053289033
800	459.3712363	0.228282033	0.032092367	463.9263015	0.221542467	0.056395533
850	540.947687	0.240874267	0.033984867	570.5733068	0.242649533	0.065540967
900	664.6876663	0.413830333	0.042068467	682.5330087	0.259049967	0.075100333
950	787.0774608	0.240746167	0.044292467	807.0324471	0.582070367	0.076869133
1000	917.0238905	0.281795767	0.047520567	941.6877418	0.317152067	0.083078367

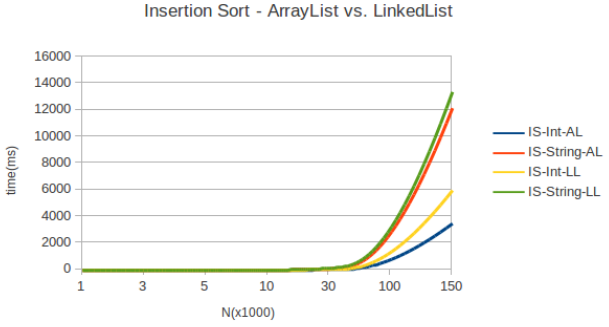
Large Values of N

For larger values of N (>10000) Merge Sort shows a clear improvement over Insertion Sort. The LinkedList data structure seems to have a slight edge over the ArrayList in Merge Sort, whereas Insertion Sort seems to prefer the ArrayList. Nevertheless, both of them leave Bubble Sort in the dust. Both the algorithms seem to handle integers better than strings in all cases.









N	MS-Int-AL	IS-Int-AL	MS-String-AL	IS-String-AL
1	0.0879481	0.064536133	0.1878103	0.0083628
3	0.466744033	0.0311809	0.347441167	0.035619933
5	1.8610181	0.250112067	1.303794367	0.4123687
10	1.9779981	0.825658133	1.6835071	0.2502449
30	5.032934333	2.8887553	7.608921767	8.0387766
100	32.67655723	80.03680457	34.51180417	398.0491204
150	100.6417746	3487.677849	181.2628985	12174.54769

N	MS-Int-LL	IS-Int-LL	MS-String-LL	IS-String-LL
1	0.225566567	0.0055802	0.0442357	0.004458433
3	1.235525433	0.027527967	0.357797867	0.035708267
5	0.1689489	0.528490033	0.218210067	0.2094561
10	0.900942667	0.158070433	1.2525347	0.232674267
30	6.066158867	2.804378633	8.481351767	7.3321917
100	22.0790233	135.3943782	41.91699867	603.9848675
150	132.5957705	5985.339675	208.0730886	13394.25507

The values of N are in the order of 1000 in the above table

Conclusion

- For small values of N(<1000), the performance is roughly : Insertion Sort > Merge Sort > Bubble Sort.
- For Larger values of N(>1000), the performance is roughly: Merge Sort >Insertion Sort > Bubble Sort.