Comp 103 Assignment 08 Results
Henry Wylde

Experiment:
The experiment done was on the efficiency of five different type of Collection classes in Java:
Array, SortedArray and LinkedList, BST and Hash.
Tests were done for the methods of adding items to each collection, removing items and checking
whether the collection contains the items. In total, three different tests were run for each set of data
(3 sets of data at counts of 20,000, 40,000 and 80,000) in order to compute an average time in
miliseconds for each bag for each different data set.
How the experiments worked to compute the times per method, is that it would determine how long
it would take to call those methods for the data counts (eg. It would check how long it takes for a
BST to add 20,000 items to it, or a LinkedList to call contains on the 20,000 data sets).

Summary:

Average totals for 20,000:	Array: 0.411
	SortedArray: 0.141
	LinkedList: 0.536
	BST: 0.0048
	Hash: 0.00095

Average totals for 40,000:	Array: 0.785
	SortedArray: 0.280
	LinkedList: 0.910
	BST: 0.0054
	Hash: 0.00073

Average totals for 80,000:	Array: 1.43
	SortedArray: 0.527
	LinkedList: 1.66
	BST: 0.0098
	Hash: 0.00176

Discussion:

For an Array:
adding is an $O(1)$ method
removing is an $O(n)$ method
contains is an $O(n)$ method.

For a SortedArray:
adding is an $O(n)$ method
removing is an $O(n)$ method
contains is an $O(\log n)$ method

For a LinkedList:
adding is an $O(1)$ method
removing is an $O(n)$ method
contains is an $O(n)$ method.

For a BST:
adding is an $O(\log n)$ method
removing is an $O(\log n)$ method
contains is an $O(\log n)$ method.

For a Hash:
adding is an $O(1)$ method
removing is an $O(1)$ method

contains is an O(1) method

For each of the different collection classes, the thearetical cost for adding/removing/containing closely matched the trial data taken from the tests. The one that didn't quite match it was a Hashset, where the time did increase as n increased, when the expected cost is thearetically a constant.

For the running the tests on the different test.txt files:
Measuring BSTBag (Test1.txt)

Data has 20000.0 items
Total: 45 milliseconds
Average: 0.00225 milliseconds

Measuring BSTBag (Test3.txt)

Data has 20000.0 items
Total: 38 milliseconds
Average: 0.0019 milliseconds

These opperations took about the same time as the random 20,000 data did, which is as expected. The reason I have not included a test for Test2.txt is because my code has some errors in it that have not been fixed. The error that displays when trying to test the Test2.txt data on the BST class is an stack overflow one. The cause is due to it getting stuck in an infinite recursion loop, where one of the nodes of the BST class has itself as a subnode further down the tree. I have not quite figured out where my error is, but I know it should be in the remove() and removeNode() sections, as that is the only place where node references are altered. I have done some testing but to no luck as of yet.

Results Table:
(Note: times are in miliseconds)

Amount of Data	Tests	Method	Bags				
			Array	SortedArray	LinkedList	BST	Hash
20000	Test 1	Adding	9	1406	2	33	14
		Contains	5584	24	5454	47	10
		Remove	2805	1399	5298	26	6
		Total	8398	2829	10754	106	30
	Test 2	Adding	12	1386	1	32	5
		Contains	5478	24	5516	36	8
		Remove	2594	1482	5490	26	5
		Total	8084	2892	11007	94	18
	Test 3	Adding	1	1387	2	22	4
		Contains	5314	34	5340	37	2
		Remove	2880	1360	5084	30	3
		Total	8195	2781	10426	89	9
	Average	Adding	0.0003667	0.06965	8.33E-005	0.0015	0.000383
		Contains	0.2729333	0.001366667	0.2718333	0.002	0.000333
		Remove	0.1379833	0.070683333	0.2645333	0.0014	0.000233
		Total	0.4112833	0.1417	0.53645	0.0048	0.00095
40000	Test 1	Adding	2	6203	16	59	12
		Contains	20287	71	18685	81	13
		Remove	11277	5265	18345	63	7
		Total	31566	11539	37046	203	32
	Test 2	Adding	5	6226	2	55	12
		Contains	20989	76	17916	69	10
		Remove	11036	5530	18352	83	7
		Total	32030	11832	36270	207	29
	Test 3	Adding	2	4855	1	64	11
		Contains	19440	58	18418	104	9
		Remove	11125	5304	17508	67	6
		Total	30567	10217	35927	235	26
	Average	Adding	0.000075	0.144033333	0.0001583	0.0015	0.000292
		Contains	0.5059667	0.001708333	0.4584917	0.0021	0.000267
		Remove	0.27865	0.134158333	0.4517083	0.0018	0.000167
		Total	0.7846917	0.2799	0.9103583	0.0054	0.000725
80000	Test 1	Adding	9	17539	6	233	67
		Contains	79531	126	72127	303	23
		Remove	51761	23269	66310	258	32
		Total	131301	40934	138443	794	122
	Test 2	Adding	5	20431	14	225	167
		Contains	73692	149	66660	293	21
		Remove	38442	22381	64243	254	30
		Total	112049	42961	130917	772	218
	Test 3	Adding	4	23079	3	237	34
		Contains	62500	146	65372	302	22
		Remove	36840	19309	64583	251	27
		Total	99344	42534	129958	790	83
	Average	Adding	0.000075	0.254370833	9.58E-005	0.0029	0.001117
		Contains	0.8988458	0.001754167	0.8506625	0.0037	0.000275
		Remove	0.5293458	0.2706625	0.8130667	0.0032	0.000371
		Total	1.4278917	0.5267875	1.663825	0.0098	0.001763