Simple Poker Game

Andres Winson 2802501123
Allenxavinzky Adjiewibowo 2802467880
L2AC

Background

During the week when the project requirements was announced, we both instantly decided to make a game of sorts, we discussed back and forth between making a card game or a mahjong game, we decided on a card game which we did another back and forth discussion whether to make Capsa or Poker, in the end with decided to make Poker as we both know the rules.

Problem Description

The problem that we want to crack from making this game is only one:

- **How do we store the deck and player hand efficiently?**

The two requirements for the data structure is:

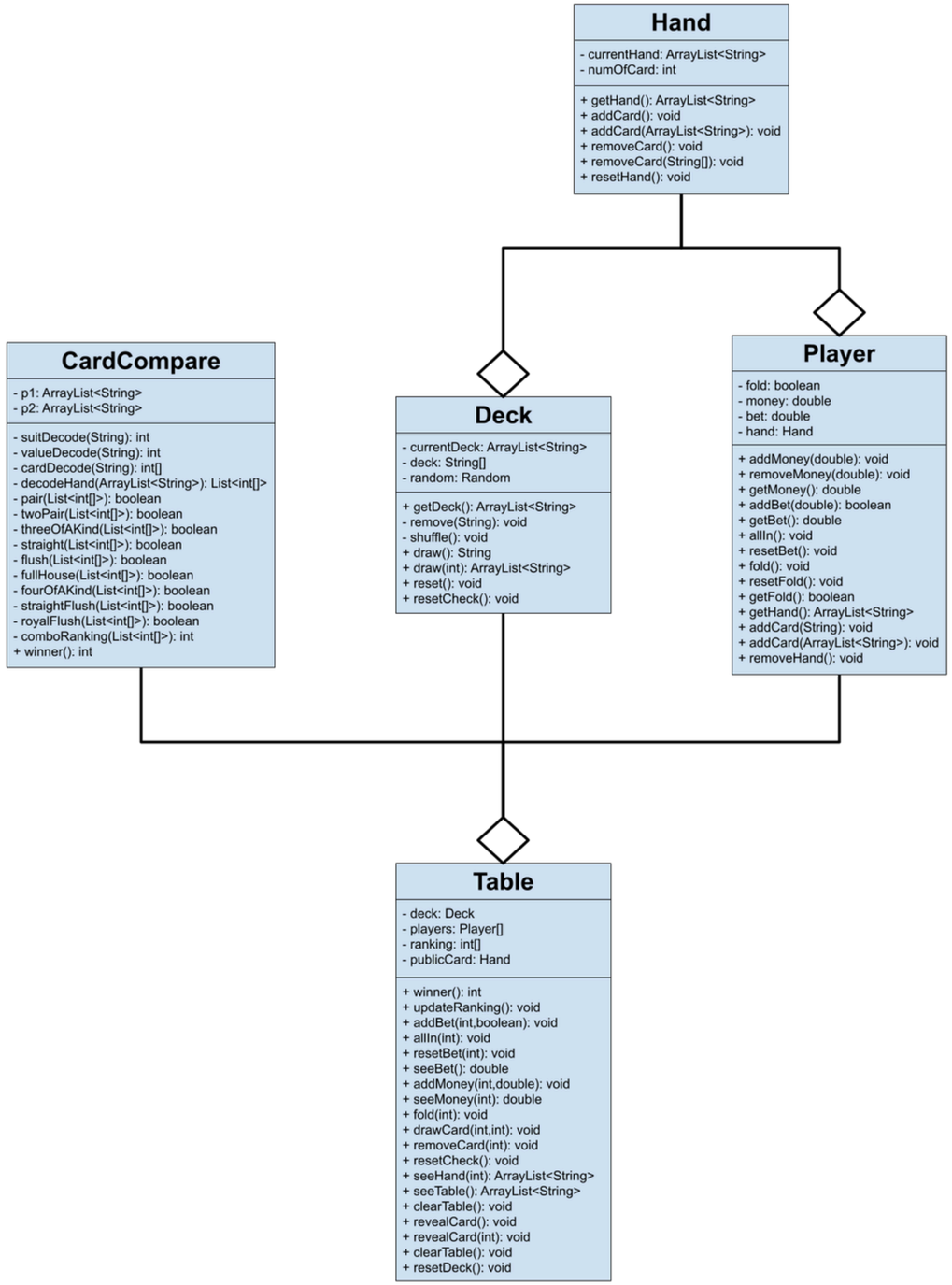
- Must be dynamic so it is scalable.
- There must be a method to randomize the elements inside

Proposed Solution

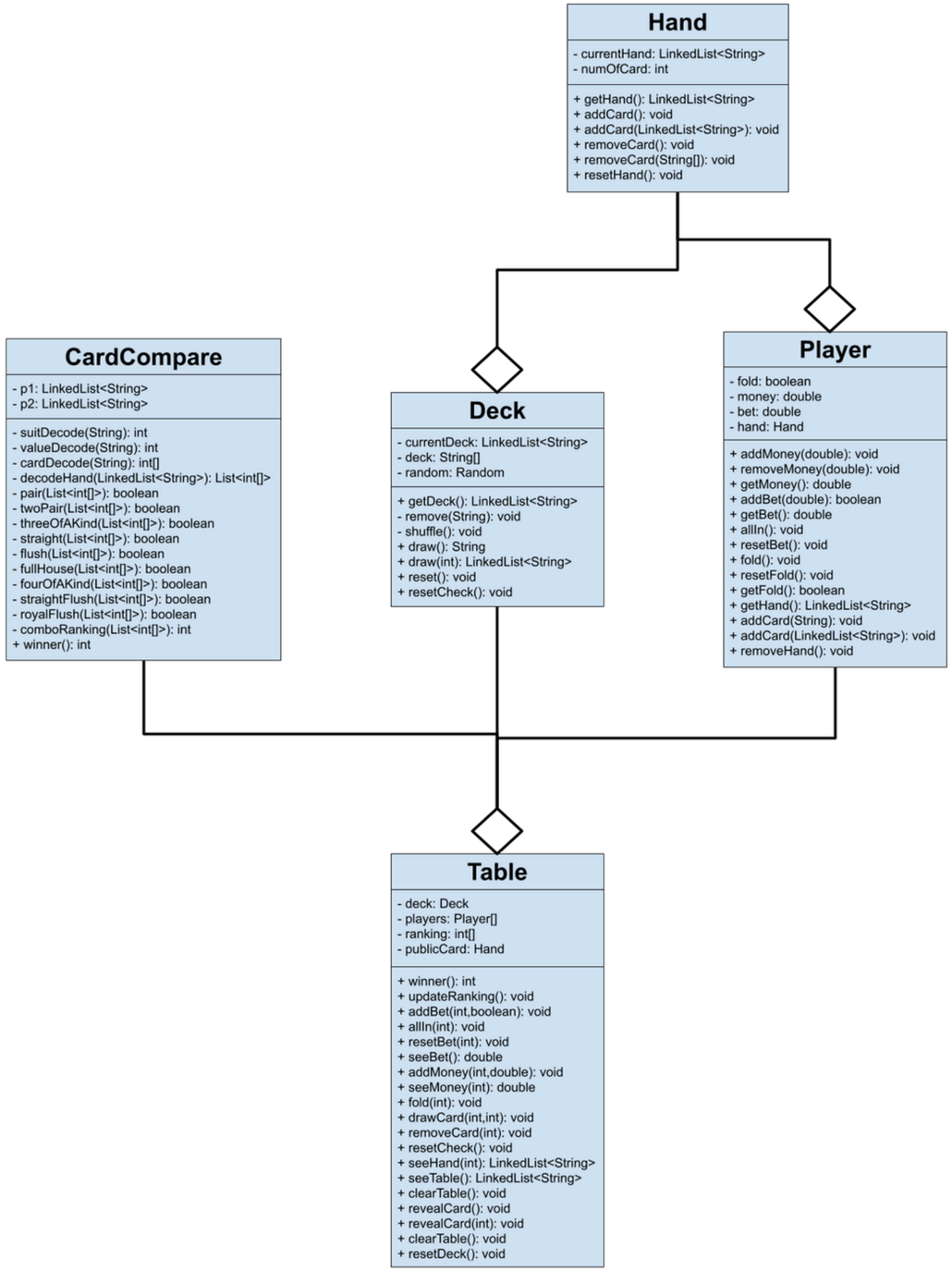
LinkedList

ArrayList

ArrayList Class Diagram



LinkedList Class Diagram



Testing

Comparison Method

- How we will test the two data structures is by comparing the memory usage and runtime of the program.

Testing Parameters:

- We will test both the memory usage and the run time of the program to determine which is more efficient

Testing Result

Poker Game Test Plan (ArrayList)												
	Trial 1		Trial 2		Trial 3		Trial 4		Trial 5		Average	
input	Total memory ussage (B)	Time (Sec)	Total memory ussage (B)	Time (Sec)	Total memory ussage (B)	Time (Sec)	Total memory ussage (B)	Time (Sec)	Total memory ussage (B)	Time (Sec)	Total memory ussage (B)	Time (Sec)
1,3,1	43896	2.930	36992	2.648	39056	2.196	36664	2.252	36688	2.395	38659.2	2.484
1,2,2,2,2,1	543776	5.569	41536	4.206	40336	3.608	40848	3.504	550984	3.978	243496.0	4.173
1,2,2,2,2,2,3,1	41040	6.738	544184	5.399	47832	5.014	43192	5.094	52024	4.469	145654.4	5.343
1,2,2,2,2,2,2,2,2,2,1	551192	7.584	42048	6.229	40824	6.720	41400	5.271	551000	6.445	245292.8	6.450
1,2,2,2,2,2,2,2,2,2,2,3,1	547144	7.560	551272	6.635	41144	7.595	48968	6.847	544416	6.391	346588.8	7.006
1,2,2,2,2,2,2,2,2,2,2,2,2,2,1	42640	9.039	41096	8.444	544752	8.338	551488	8.037	41088	8.562	244212.8	8.484
1,2,2,2,2,2,2,2,2,2,2,2,2,2,3,1	544552	10.381	544872	8.678	544488	10.181	41896	9.003	42400	8.104	343641.6	9.269
1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1	544488	11.125	41160	8.428	42552	10.317	41152	10.787	42344	9.307	142339.2	9.993
1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3,1	547248	10.175	545704	9.584	42344	11.207	547840	9.516	42880	9.102	345203.2	9.917
1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1	43544	11.314	42520	9.750	42192	10.269	546144	9.790	42480	10.004	143376.0	10.225

Poker Game Test Plan (LinkedList)												
	Trial 1		Trial 2		Trial 3		Trial 4		Trial 5		Average	
input	Total memory ussage (B)	Time (Sec)	Total memory ussage (B)	Time (Sec)	Total memory ussage (B)	Time (Sec)	Total memory ussage (B)	Time (Sec)	Total memory ussage (B)	Time (Sec)	Total memo	Time (Sec)
1,3,1	37840	3.163	541200	2.545	37864	3.450	38184	2.921	543504	2.725	239718.4	2.961
1,2,2,2,2,1	42888	4.276	42472	4.794	545768	4.613	42432	4.144	43224	4.297	143356.8	4.425
1,2,2,2,2,2,3,1	552168	6.811	545216	5.371	545112	5.433	41976	5.738	48672	5.139	346628.8	5.698
1,2,2,2,2,2,2,2,2,2,1	43040	6.785	551472	6.056	50696	6.570	43688	6.248	553504	5.664	248480.0	6.265
1,2,2,2,2,2,2,2,2,2,3,1	545656	7.858	545792	7.336	45080	7.415	49032	7.590	553424	7.326	347796.8	7.505
1,2,2,2,2,2,2,2,2,2,2,2,1	546944	8.666	42160	6.842	42496	8.067	547360	8.241	553576	8.349	346507.2	8.033
1,2,2,2,2,2,2,2,2,2,2,2,3,1	43008	9.246	43368	8.596	545944	9.491	42608	9.125	551120	9.254	245209.6	9.142
1,2,2,2,2,2,2,2,2,2,2,2,2,2,1	547624	8.698	43024	8.892	548688	11.128	547816	11.243	44808	11.606	346392.0	10.313
1,2,2,2,2,2,2,2,2,2,2,2,2,2,3,1	44904	11.675	43736	10.033	45416	10.149	43504	8.932	547056	8.124	245408.0	9.783
1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1	547328	10.403	44480	10.496	547400	10.741	546992	11.259	548400	10.647	446920.0	10.709

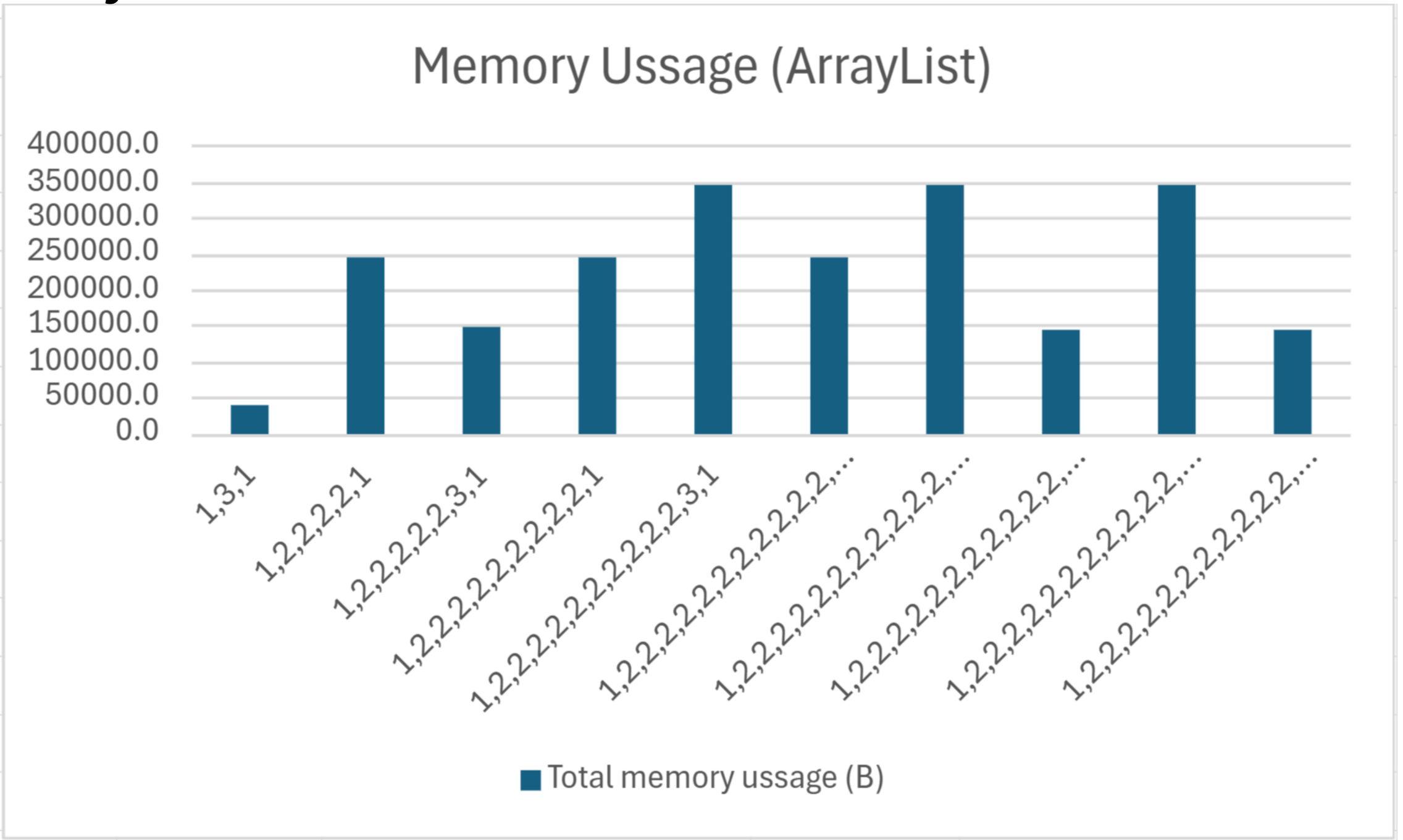
ArrayList

input	Average	
	Total memory usage (B)	Time (Sec)
1,3,1	38659.2	2.484
1,2,2,2,2,1	243496.0	4.173
1,2,2,2,2,2,3,1	145654.4	5.343
1,2,2,2,2,2,2,2,2,2,1	245292.8	6.450
1,2,2,2,2,2,2,2,2,2,2,3,1	346588.8	7.006
1,2,2,2,2,2,2,2,2,2,2,2,2,2,1	244212.8	8.484
1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3,1	343641.6	9.269
1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1	142339.2	9.993
1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3,1	345203.2	9.917
1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1	143376.0	10.225

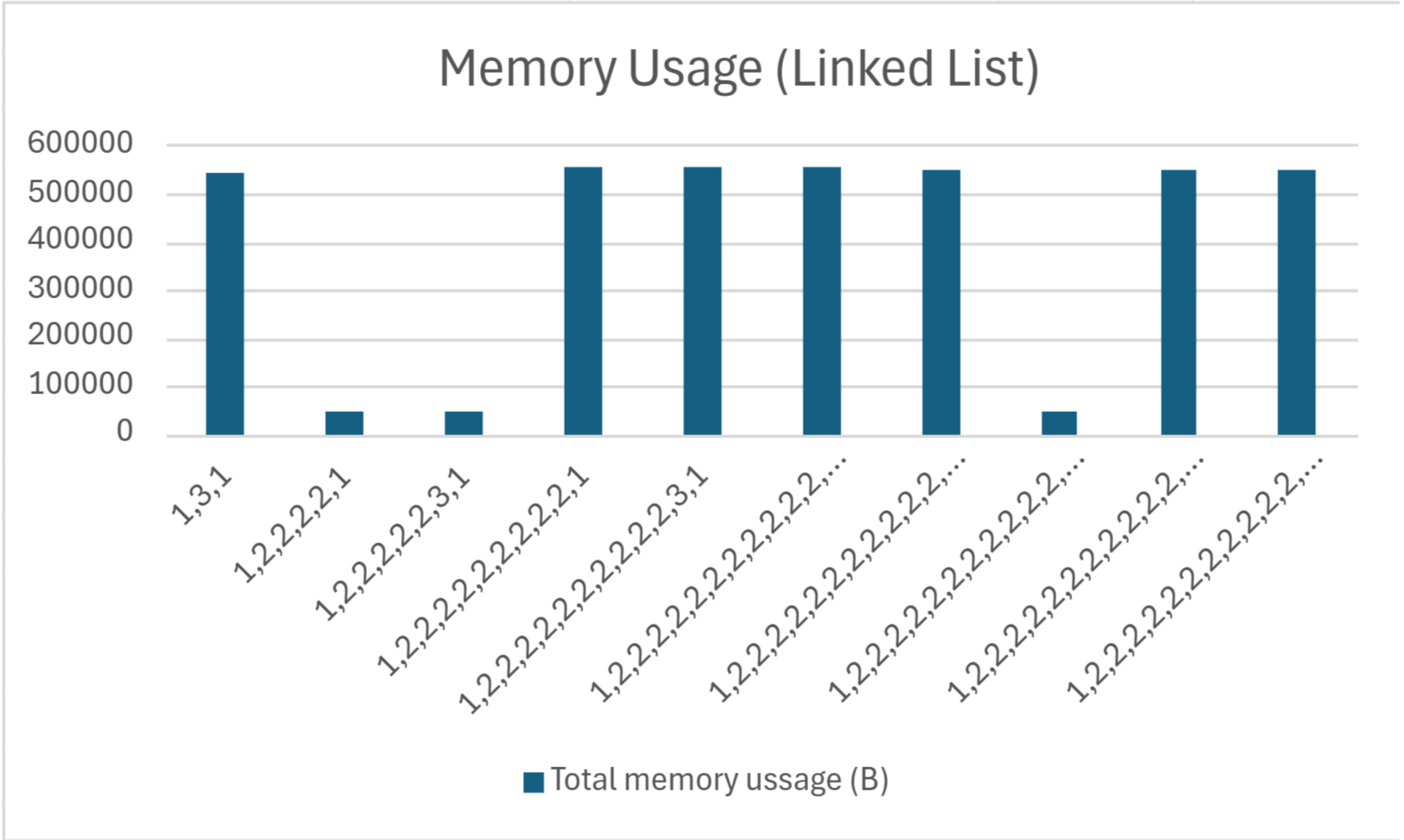
LinkedList

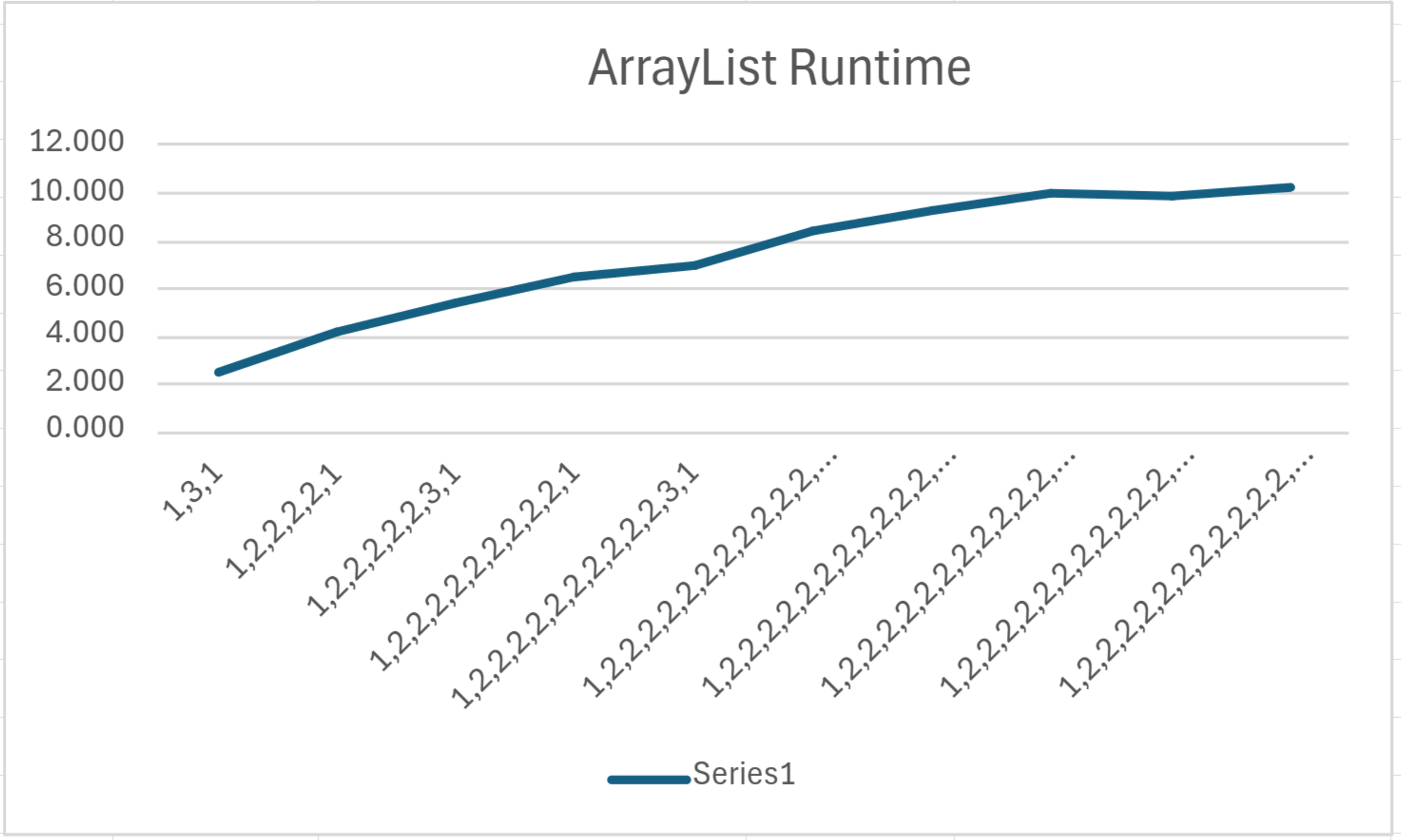
input	Average	
	Total memory usage (B)	Time (Sec)
1,3,1	239718.4	2.961
1,2,2,2,2,1	143356.8	4.425
1,2,2,2,2,2,3,1	346628.8	5.698
1,2,2,2,2,2,2,2,2,2,1	248480	6.265
1,2,2,2,2,2,2,2,2,2,2,3,1	347796.8	7.505
1,2,2,2,2,2,2,2,2,2,2,2,2,2,1	346507.2	8.033
1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3,1	245209.6	9.142
1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1	346392	10.313
1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3,1	245408	9.783
1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1	446920	10.709

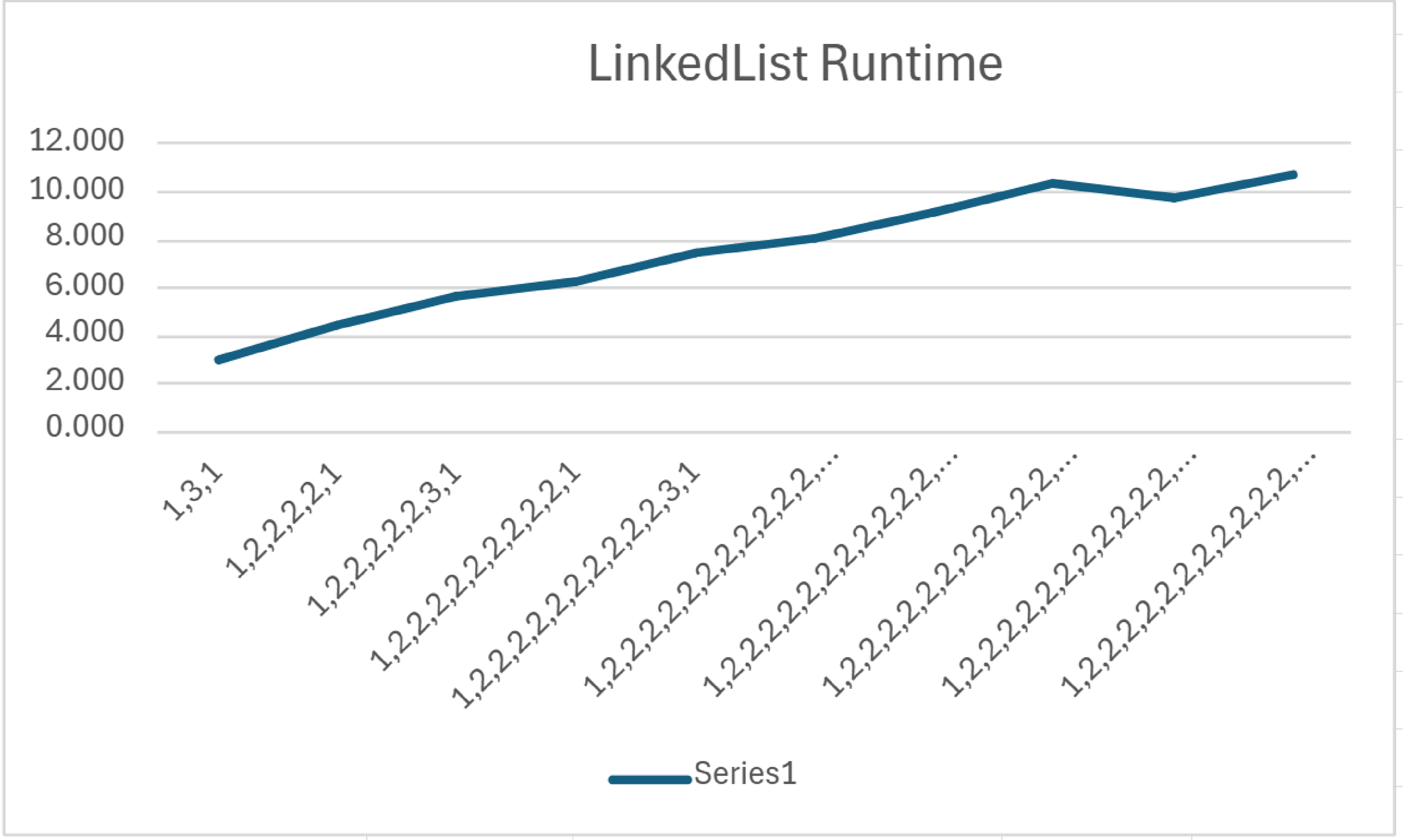
ArrayList


Memory Ussage (ArrayList)
400000.0
350000.0
300000.0
250000.0
200000.0
150000.0
100000.0
50000.0
0.0
1,3,1
1,2,2,2,2,1
1,2,2,2,2,3,1
1,2,2,2,2,2,2,2,2,1
1,2,2,2,2,2,2,2,2,2,3,1
1,2,2,2,2,2,2,2,2,2,2,...
1,2,2,2,2,2,2,2,2,2,2,...
1,2,2,2,2,2,2,2,2,2,2,...
1,2,2,2,2,2,2,2,2,2,2,...
1,2,2,2,2,2,2,2,2,2,2,...
Total memory ussage (B)

LinkedList


Memory Usage (Linked List)
600000
500000
400000
300000
200000
100000
0
1,3,1
1,2,2,2,2,1
1,2,2,2,2,3,1
1,2,2,2,2,2,2,2,2,1
1,2,2,2,2,2,2,2,2,3,1
1,2,2,2,2,2,2,2,2,2,...
1,2,2,2,2,2,2,2,2,2,...
1,2,2,2,2,2,2,2,2,2,...
1,2,2,2,2,2,2,2,2,2,...
1,2,2,2,2,2,2,2,2,2,...
Total memory ussage (B)


ArrayList Runtime
12.000
10.000
8.000
6.000
4.000
2.000
0.000
1,3,1
1,2,2,2,2,1
1,2,2,2,2,3,1
1,2,2,2,2,2,2,2,1
1,2,2,2,2,2,2,2,3,1
1,2,2,2,2,2,2,2,2,2,...
1,2,2,2,2,2,2,2,2,2,...
1,2,2,2,2,2,2,2,2,2,...
1,2,2,2,2,2,2,2,2,2,...
1,2,2,2,2,2,2,2,2,2,...
Series1


LinkedList Runtime
12.000
10.000
8.000
6.000
4.000
2.000
0.000
1,3,1
1,2,2,2,2,1
1,2,2,2,2,3,1
1,2,2,2,2,2,2,2,1
1,2,2,2,2,2,2,2,2,3,1
1,2,2,2,2,2,2,2,2,2,2,...
1,2,2,2,2,2,2,2,2,2,2,...
1,2,2,2,2,2,2,2,2,2,2,...
1,2,2,2,2,2,2,2,2,2,2,...
1,2,2,2,2,2,2,2,2,2,2,...
Series1

Conclusion

	ArrayList	LinkedList
AVG	223846.4	295641.760
TOTAL	2238464.0	2956417.6
	ArrayList	LinkedList
Runtime	7.334	7.483

ArrayList is the winner

Program Showcase