Konstantin Zelmanovich
Dr. Krzysztof Nowak
CS 260 - 004
Programming Assignment 1
February 19, 2018

Timing Results

After completing all the timing test cases, here are the conclusions:

List:

1. **Iterated Insertion Forward** – very efficient and fast procedure, no matter what implementation type is used, even though, it takes longer time to complete for arrays. Built-in implementation is the fastest, followed by the pointer implementation.
2. **Iterated Insertion Backward** – same thing as the Forward
3. **Traversal** – built-in python implementation is the fastest, followed by the array implementation and pointer implementation is the slowest, therefore less efficient for pointers. This can be explained by the fact that temporary pointers have to be set in order to traverse through the list.
4. **Iterated Deletion Forward** – built-in python implementation is the fastest, followed by the array implementation and the slowest is the pointer implementation.
5. **Iterated Deletion Backward** – same as Forward

Stack:

1. **Iterated Insertion PUSH operation** – the fastest is the built-in implementation, followed by the pointer implementation and the array implementation is the slowest. This can be explained that the pointer implementation is faster than the array implementation because stack deals with just the first head) of the structure.
2. **Iterated Deletion POP operation** – the fastest is the pointer implementation and built-in and array implementations are about the same, even though an array is slightly more efficient

Next 2 pages have all the tables with the timing results.

List Testing Iterated Insertion - Forward			
Data Structure Size (# of elements)	**Built-in (ms)**	**Array (ms)**	**Pointer (ms)**
1000	0.53	3.6	1.27
1500	1.01	7.08	1.97
2000	1.7	12.48	2.49
2500	2.59	18.64	3.17

List Testing Iterated Insertion - Backward			
Data Structure Size (# of elements)	**Built-in (ms)**	**Array (ms)**	**Pointer (ms)**
1000	0.12	2.84	4.33
1500	0.16	5.54	6.34
2000	0.26	9.71	8.47
2500	0.27	14.37	10.31

List Testing Traversal			
Data Structure Size (# of elements)	**Built-in (ms)**	**Array (ms)**	**Pointer (ms)**
1000	0.04	0.54	656.9
1500	0.07	0.58	1480.22
2000	0.09	0.8	2685.12
2500	0.13	1.02	3908.1

List Testing Iterated Deletion - Forward			
Data Structure Size (# of elements)	**Built-in (ms)**	**Array (ms)**	**Pointer (ms)**
1000	2.37	4.36	527.97
1500	5.34	10.23	1217.96
2000	9.18	19.4	2110.88
2500	15.52	25.98	3406.99

List Testing Iterated Deletion - Backward			
Data Structure Size (# of elements)	**Built-in (ms)**	**Array (ms)**	**Pointer (ms)**
1000	0.14	4.09	935.63
1500	0.19	8.07	2083.54
2000	0.26	14.43	3532.25
2500	0.37	25.39	5041.73

Stack Testing Iterated Insertion - PUSH operation			
Data Structure Size (# of elements)	**Built-in (ms)**	**Array (ms)**	**Pointer (ms)**
1000	0.37	51.24	0.92
1500	0.79	118.85	1.43
2000	0.67	205.08	2.95
2500	0.92	340.73	2.68

Stack Testing Iterated Deletion - POP operation			
Data Structure Size (# of elements)	**Built-in (ms)**	**Array (ms)**	**Pointer (ms)**
1000	0.97	0.75	0.78
1500	1.4	1.04	1.06
2000	2.85	2.01	1.87
2500	3.04	2.05	1.77