

*I performed two comparisons, based on compiling in both home and flip environments. I did this because I was unsatisfied with the values that I received from flip, when looping with relatively small numbers.

Problem 1: Bag implementation using LinkedList

Compilation source: Home			Flip		
	Memory Used			Memory Used	
	(kb)	Time (ms)		(kb)	Time (ms)
1000	32,768.0	3.25	1000	-	-
2000	65,536.0	9.64	2000	-	-
4000	126,976.0	39.14	4000	-	30.00
8000	262,144.0	152.12	8000	-	120.00
16000	520,192.0	541.79	16000	20.0	510.00
32000	1,040,384.0	2,266.97	32000	516.0	2,030.00
64000	2,080,768.0	9,027.69	64000	1,520.0	8,120.00
128000	4,169,728.0	38,565.70	128000	3,520.0	32,430.00
256000	8,335,360.0	173,258.00	256000	7,520.0	218,970.00

Problem 2: Bag implementation using DynArr

Compilation source: Home			Flip		
	Memory Used			Memory Used	
	(kb)	Time (ms)		(kb)	Time (ms)
1000	12,288.0	2.03	1000	-	-
2000	28,672.0	7.64	2000	-	-
4000	57,344.0	25.42	4000	-	20.00
8000	122,880.0	95.63	8000	-	90.00
16000	249,856.0	381.44	16000	-	380.00
32000	512,000.0	1,490.85	32000	-	1,530.00
64000	1,024,000.0	6,008.02	64000	160.0	6,140.00
128000	2,048,000.0	23,212.20	128000	664.0	24,560.00
256000	4,096,000.0	96,118.80	256000	1,660.0	98,380.00

Questions:

1

The linkedList implementation uses more memory. A Linked List occupies so much more space

because each pointer requires extra memory space for each element of a list. This is because unlike an array, a linked list does not store values sequentially, and therefore cannot index easily, nor efficiently. While this provides other potential benefits, memory efficiency is not one.

2

An array list is much faster than a linked list. The reason is because indexing is done in constant time, compared to linear time, $O(n)$, for linked lists.

3

The time complexity for a linked list is linear when accessing/searching a list. When deleting from a linked list, time complexity moves to constant as no re-size would be necessary. For a dynamic array, accessing a value is constant, but deleting a value moves to linear. With the values above, I show that linked lists are 2x slower than dynamic arrays. Deleting is one of the few areas where linked lists shine and therefore, I think the results would be much closer, had the called function been remove, rather than contains.