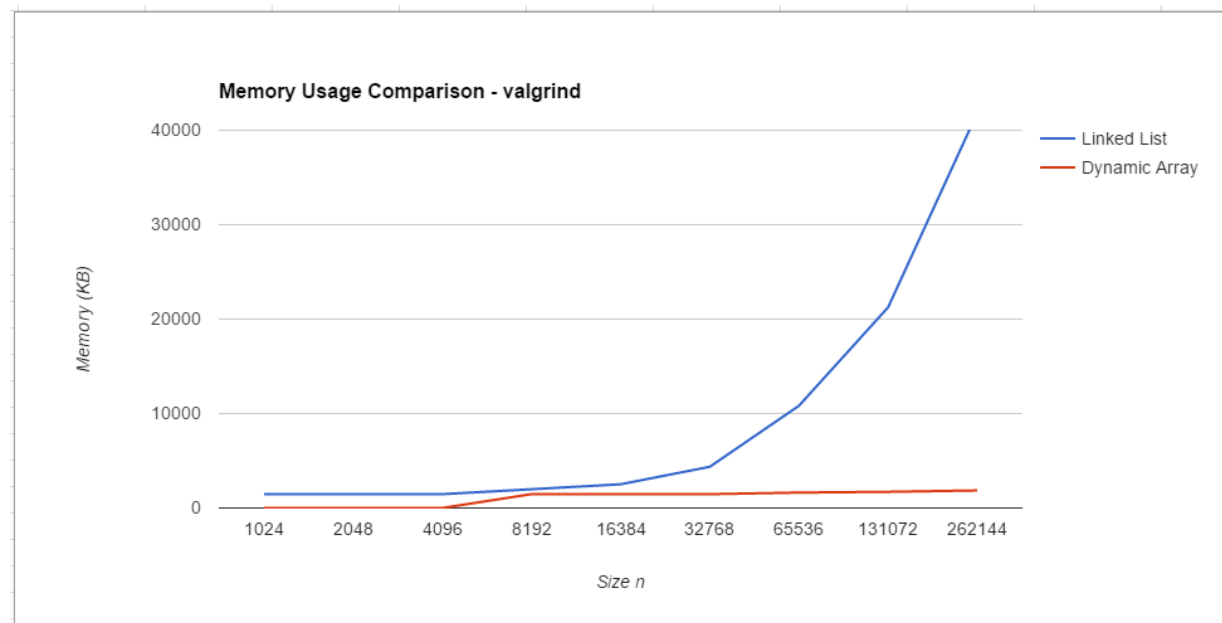
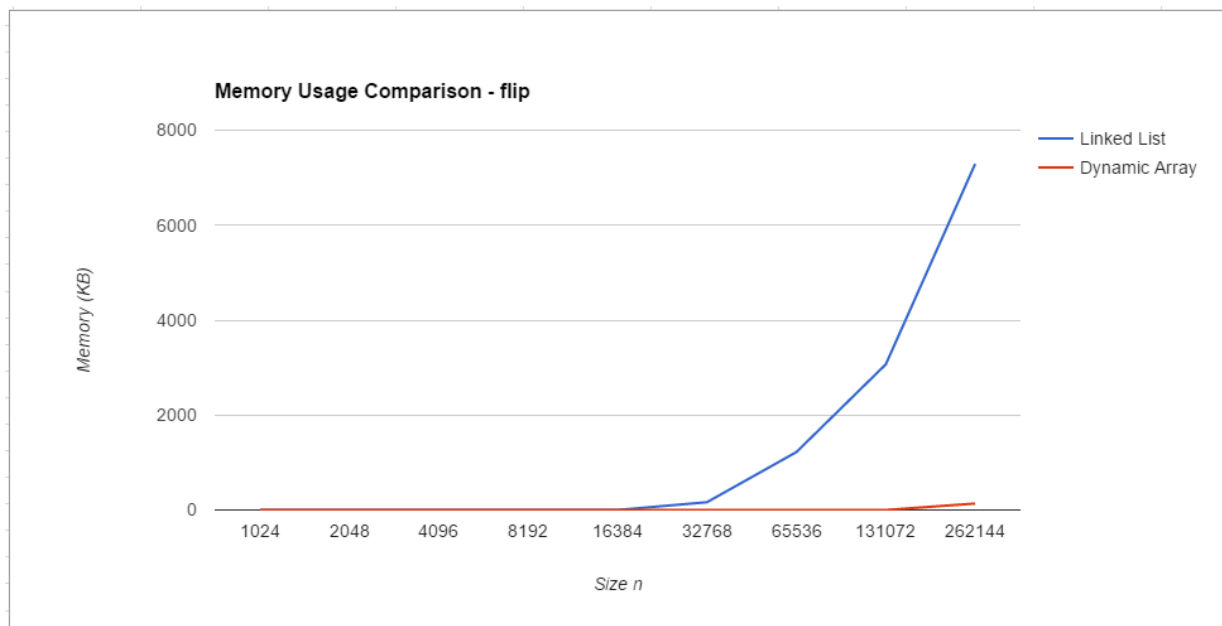


Size n	Execution Time (ms)			
	Linked List		Dynamic Array	
	flip	valgrind	flip	valgrind
1024	0	50	0	20
2048	0	150	0	80
4096	30	470	20	310
8192	140	1230	110	1280
16384	590	4860	440	4950
32768	2350	19660	1790	20050
65536	9520	77430	7500	79340
131072	38530	320290	28830	308760
262144	155060	1247410	115450	1231310



Size n	Memory Usage (KB)			
	Linked List		Dynamic Array	
	flip	valgrind	flip	valgrind
1024	0	1464	0	0
2048	0	1464	0	0
4096	0	1464	0	0
8192	0	1984	0	1464
16384	0	2512	0	1464
32768	164	4360	0	1464
65536	1220	10824	0	1632
131072	3068	21248	0	1712
262144	7292	41832	136	1856

1. Which of the implementations uses more memory? Explain why.

The linked list implementation uses significantly more memory. This is due to the linked list storing the locations of the next and previous link whereas in an array the memory is stored contiguously and these pointers are not required.

2. Which of the implementations is the fastest? Explain why.

The dynamic array implementation is fastest because the dynamic array's memory is stored contiguously and can be iterated through faster than the data of the linked list which is not stored contiguously.

3. Would you expect anything to change if the loop performed `remove()` instead of `contains()`? If so, why?

I would expect the execution time of the dynamic array implementation to be much longer. In a linked list, removing an item (once found) is just a matter of adjusting pointers and freeing the memory. In the dynamic array, once the item is found, each element after it must be shifted over.