

The sequence generation function has a space complexity of $O((\log n)^2)$. There is one for loop nested within another for loop. The nested for loop repeatedly runs one time more than the previous iteration. Hence the space complexity is $(\log n)^2$.

For space complexity of this function is equal to $O(n)$, because we malloc-ed a linked list that points to each sub-array. This will be a minimum of $n/2$ and maximum of n , hence the space complexity is $O(n)$.

Since $O(n) + O((\log n)^2) = O(n)$, **the overall (worst case) space complexity is $O(n)$.**

Shell Insertion sort		
Input Size	IO Time (sec)	Sorting Time (sec)
15	0	0
1,000	0	0
10,000	0	.03
100,000	0	1.08
1,000,00	0.17	58

The time complexity of shell sort is $O(n \log n)$ (because of the shell sort) best case and $O(n^2)$ (because of insertion sort) is worst case. Hence, $O(n \log n) \leq O(x) \leq O(n^2)$. Hence, the IO run time and sorting time increases with n .

The IO run time takes approximately the same time as the previous assignment (I used fread in both). The sorting time is

slower in the linked list version of shell insertion sort because we had to create a linked list that points to the sub array linked lists. In addition to this, the swapping could not be done in one step as it was done in the array version of this assignment.