ECE 368 – PA01 – Karim Itani

The sequence generation function has a time complexity of O(n(logn)^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 big O is n(logn)^2.

For space complexity of this function is equal to O(n), because we malloc-ed n to the array that contains the sub array sequence. Even though the array will not every need this much space, it ensures that the array never overflows.

| **Shell Insertion sort** | | | |
|---|---|---|---|
| **Input Size** | **Time (sec)** | **comparisons** | **moves** |
| 15 | 0 | 93 | 15 |
| 1,000 | 0 | 35730 | 4543 |
| 10,000 | 0 | 623628 | 71109 |
| 100,000 | 0 | 9611932 | 1004825 |
| 1,000,00 | 2 | 137680165 | 13479446 |

| **Shell Selection sort** | | | |
|---|---|---|---|
| **Input Size** | **Time (sec)** | **comparisons** | **moves** |
| 15 | 0 | 473 | 45 |
| 1,000 | 0 | 1554965 | 13629 |
| 10,000 | 0 | 150943093 | 213327 |
| 100,000 | 22 | 15014392012 | 3014475 |
| 1,000,00 | N/A | N/A | N/A |

As we can see from the table above, as the number of inputs grew, the time taken to sort the numbers grew at an increasing rate.

The shell insertion sort has a time complexity equal to O(n^2/log(n)) and a space complexity equal to O(1). No space is allocated in this function besides that allocated in the function call to the sequence generation function.

The shell selection sort has a time complexity equal to O(n^2) and a space complexity equal to O(1). No space is allocated in this function besides that allocated in the function call to the sequence generation function.