Siddharth Prabhu

ECE 368

PA01 Report

An analysis of the time- and space-complexity of your algorithm to generate the sequence (not sorting).

The algorithm used to generate the sequence had a run time complexity of O(n^2) as a nested for loop was used to generate the sequence. The space complexity was O(n) as no recursive call was used but memory was allocated on the heap.

• A tabulation of the run-time, number of comparisons, and number of moves obtained from running your code on some sample input files. You should comment on how the run-time, number of comparisons, and number of moves grow as the problem size increases, i.e., the time complexity of your routines.

The running time of shell insertion sort and Shell selection sort are shown in the table below.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sorting Size | Shell Insertion Sort(seconds) | Shell Selection Sort (seconds) | Moves  Insertion | Comparisons  Insertion | Moves Selection | Comparison Selection |
| 15 | 0.00 | 0.00 | 75 | 90 | 45 | 353 |
| 1000 | 0.00 | 0.00 | 30955 | 35498 | 4543 | 1514965 |
| 10000 | 0.00 | 0.23 | 550711 | 621820 | 71109 | 150273093 |
| 100000 | 0.20 | 30.16 | 8605411 | 9610236 | 1004825 | 15004292012 |
| 1000000 | 2.94 | 3521.17 | 123987151 | 137466597 | 40438338 | 284000000 |

As the problem size grew the run time also increased as evidenced from the table above. The time complexity of the sorting varied between the two as the insertion sort was more efficient than selection sort on arrays where already partially sorted. For the larger sample sizes, the insertion sort became very efficient once the arrays started to sort themselves whereas the selection sort has the same complexity for both best case and worst case situations of O(n^2) making it very slow especially for the one million sample size.

• A summary of the space complexity of your sorting routines, i.e., the complexity of the additional memory required by your routines.

The shell insertion sort had a space complexity of O(n) due to memory allocated on the heap to store the array. No further memory was allocated during the sorting process. So, the sorting process only had a time complexity of O(n^2). Shell selection sort also had a time complexity of O(n^2) but as explained above it was much slower than the insertion sort. The shell selection sort also had a space complexity of O(1).