Austin Hoffmann
ECE 368
02/03/2017

Programming Assignment 1 Write-up

Time Analysis of sequence generator:
The first computationally intensive task the algorithm has to perform is to find the maximum power to which 2 can be raised, so that 2 to that power is greater than the size of the array. This operation has a time complexity of log2(n), where n is the size of the array. This implies that the second loop has a time complexity of log2(n) * log2(n), giving the overall algorithm a final time complexity of O(log2(x^2)).
Space Analysis of sequence generator:
This algorithm only has one place at which the amount of memory required to perform the algorithm, and that is where it malloc()'s the actual sequence of numbers. This value grows linearly as the size of the array increases, so the algorithm is O(n).

Tabulation of Results:

Insertion Sort

FileSize	1000	10000	100000	1000000
Int Read	1000	10000	100000	1000000
Int Stored	1000	10000	100000	1000000
Sequence Len	40	67	101	142
Comparisons	9.086 e+03	1.42218e+05	2.009e+06	2.696e+07
Moves	6.6543e+04	1.1725e+06	1.8215e+07	2.614e+08
I/O Time	2.74e-04	3.05e-04	8.97e-04	9.199e-03
Sorting Time	1.23e-04	1.751e-03	2.5167e-02	3.534e-01

Time Complexity: O(nlog^2(n))

Selection Sort

FileSize	1000	10000	100000	1000000
Int Read	1000	10000	100000	NULL
Int Stored	1000	10000	100000	NULL
Sequence Len	40	67	101	NULL
Comparisons	1.44e+06	1.49e+08	1.498e+10	NULL
Moves	2.126e+04	3.24e+05	4.546e+06	NULL
I/O Time	9.7e-05	1.68e-04	6.56e-04	NULL
Sorting Time	1.836e-03	1.565e-01	1.870994e+01	NULL

Time Complexity: O(n^2)
Selection sort is much more inefficient due to the order of magnitude of comparisons being so much higher than Insertion sort.

The space complexity of the sorting algorithm depends linearly on both the size of the sequence, and the size of the array. Since the size of the sequence of depends on the log2(n), the total space complexity can be estimated to be O(nlog2(n)).