**Bangladesh University of Engineering and Technology**

**Department of Computer Science and Engineering**

**CSE 204**

**Data Structures and Algorithms I Sessional**

Offline 2.1

Title: Running Time Comparison Between Selection Sort and Insertion Sort in Average Case, Best Case and Worst Case

Offline 2.2

Title: Running Time Comparison and Complexity Analysis of Power Set Generator Algorithm.

Submitted by:

Name: Tanzim Hossain Romel

Student ID: 1705069

Dept: CSE

Sec: B-1

Date of Submission: 09/06/2019

**Machine Configuration:**

Processor: Intel Core i5 8300H CPU @ 2.30GHz, 2304 MHz, 4 Cores, 8 Logical Processors

OS Name: Microsoft Windows 10 Home

Total Physical Memory: 15.9 GB

Total Virtual Memory: 20.5 GB

GPU: NVIDIA GeForce GTX 1050Ti (4GB)

**Selection Sort:**

The selection sort algorithm sorts an array by repeatedly finding the minimum element from unsorted part and putting it at the beginning.

For best case scenario (when the array is already sorted) selection sort doesn’t swap any element with the unsorted part for minimum value. For worst case scenario (when the array is sorted in descending order) it has to swap with every element from unsorted part for minimum value. Though for both cases time complexity will be O(n^2) as swapping is done in constant time. So the time difference is negligible among best, average and worst case scenario.

Time Complexity: O(n^2) as there are two nested loops.

Space Complexity: O(1)

The table & graph is attached herewith.

|  |  |  |  |
| --- | --- | --- | --- |
| Input Size | Best Case (ns) | Average Case (ns) | Worst Case (ns) |
| 10 | 99.8 | 598.7 | 498.4 |
| 100 | 8575.1 | 26329.5 | 34898.4 |
| 200 | 35398.3 | 108420 | 129558 |
| 500 | 227309 | 611009 | 807268 |
| 1000 | 899194 | 2.51E+06 | 3.21E+06 |
| 2000 | 3.56E+06 | 1.10E+07 | 1.28E+07 |
| 3000 | 8.53E+06 | 2.92E+07 | 3.20E+07 |
| 5000 | 2.26E+07 | 8.10E+07 | 8.38E+07 |
| 7000 | 4.51E+07 | 1.65E+08 | 1.66E+08 |
| 9000 | 7.27E+07 | 2.69E+08 | 2.64E+08 |
| 10000 | 8.97E+07 | 3.34E+08 | 3.26E+08 |

**Insertion Sort:** Insertion sort runs in O(n) time in its best case and runs in O(n^2) in its worst and average cases.

Insertion sort performs two operations: it scans through the list, comparing each pair of elements, and it swaps elements if they are out of order. Each operation contributes to the running time of the algorithm. If the input array is already sorted, insertion sort compares elements and performs no swaps. Therefore, in the best case, insertion sort runs in O(n) time. But for average and worst-case scenario the inner loop performs some swaps and the running time becomes O(n^2).

Time Complexity: O(n^2) for worst case and O(n) for best case.

Space Complexity: O(1)

The table & graph is attached herewith.

|  |  |  |  |
| --- | --- | --- | --- |
| Input Size | Best Case (ns) | Average Case (ns) | Worst Case (ns) |
| 10 | 0 | 99.8 | 299.2 |
| 100 | 199.3 | 8977.2 | 10681.1 |
| 200 | 997.1 | 23538.7 | 41576.6 |
| 500 | 1398.7 | 142801 | 274166 |
| 1000 | 2196.2 | 550884 | 1.09E+06 |
| 2000 | 5.03E+03 | 2.19E+06 | 4.32E+06 |
| 3000 | 1.19E+04 | 5.31E+06 | 1.07E+07 |
| 5000 | 1.99E+04 | 1.41E+07 | 2.84E+07 |
| 7000 | 2.69E+04 | 2.78E+07 | 5.48E+07 |
| 9000 | 3.00E+04 | 4.49E+07 | 8.89E+07 |
| 10000 | 3.39E+04 | 5.49E+07 | 1.10E+08 |

**Power Set Generator Algorithm:** Power set P(S) of a set S is the set of all subsets of S. If S has n elements in it then P(S) will have 2^n elements.

So, to find all subsets the time complexity will be O(2^n) and to print a subset the time complexity will be O(n). So to find and print all the subset the time complexity is O(n\* n^2 ).

This is an exponential algorithm. So, the runtime grows even faster than polynomial algorithm based on n. We see from the table as the number of elements increase the runtime increases greatly.

The data table is attached herewith with the graph:

|  |  |
| --- | --- |
| Input Size | Running Time (ns) |
| 5 | 0 |
| 10 | 0 |
| 15 | 997000 |
| 20 | 24932000 |
| 25 | 1004360000 |
| 30 | 39135313000 |
| 35 | 1.4783E+12 |

**Comparison Between Best Case, Average Case, Worst Case:**

Best Case:

Average Case:

Worst Case: