# **Assignment** **6: Medians and Order Statistics & Elementary Data Structures**

Jacob Jeffers

MSCS 532

Dr. Vanessa Cooper

Fall 2024

11/24/2024

Assignment 6: Medians and Order Statistics & Elementary Data Structures

Part 1: Implementation and Analysis of Selection Algorithms

For the first part of this assignment, I needed to implement the deterministic selection algorithm in VS Code. To do this, I divided the array into groups of 5, found the median of each group, then recursively found the median of those medians to use as the pivot number, partition an array around the median, and the search that for the smallest element. The time complexity is O(n) for the worst case, while the space is 0(log n) (Cormen et al., 2022). Implementing this in VS Code took some practice, but we have been in VS Code all term, and I feel like I am able to do things a lot faster than before. A screenshot of the implemented code can be found below; I will include the repository as well at the bottom of the assignment.

A screen shot of a computer program

Description automatically generated

The next part of the assignment involved implementing the randomized selection algorithm. To do this, I selected a pivot at random, partitioned an array around that pivot, and recursively searched the partition containing the smallest element. The time complexity is O(n^2) for the worst-case scenario and the space complexity is O(log n) (Cormen et al., 2022). A screenshot of the implemented code can be found below.

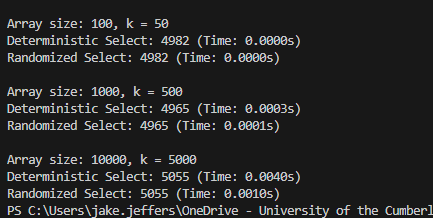
A screen shot of a computer program

Description automatically generated

For the empirical analysis, I generated different sized arrays with different distributions (random, sorted, reverse sorted, and arrays with duplicates). From there, I measured the running time of both algorithms to find the smallest element. A screenshot of the code can be found below as well as the output.

A screen shot of a computer code

Description automatically generated



Empirically, both algorithms were successful at finding the smallest element. The randomized select algorithm performed faster at larger array sizes due to the deterministic algorithm having the overhead of finding medians. The randomized select algorithm seems more practical for general use cases, but the deterministic algorithm benefits from consistency which would be excellent in applications where worst-case performance guarantees are critical.

Part 2: Elementary Data Structures Implementation and Discussion

To begin this part, I implemented each part individually with different .py files. For example, I implemented the arrays as one file, matrices as another, and so on. This made running and testing the code much easier. The screenshot below lists each file for this assignment.

A screenshot of a computer program

Description automatically generated

The table below illustrates the time complexity for each data structure implemented in VS Code.

#### **Time Complexity of Operations (Cormen et al., 2022)**

| **Data Structure** | **Operation** | **Time Complexity** |
| --- | --- | --- |
| Array | Insert/Delete | O(n) |
|  | Access | O(1) |
| Stack | Push/Pop | O(1) |
|  | Peek | O(1) |
| Queue | Enqueue/Dequeue | O(1) |
| Linked List | Insert/Delete | O(n) (worst) |
|  | Traverse | O(n) |

Arrays offer O(1) time complexity for access operations but O(n) for insertion and deletion, making them ideal for static datasets. Matrices are useful for multidimensional data manipulation, with operations like row deletion requiring O(n). Stacks and Queues demonstrate efficient O(1) operations for push/pop and enqueue/dequeue. Linked Lists provide efficient insertion and deletion, especially when the position is known, but traversal is O(n).

The test code can be found fully in the repository as well as instructions for running each code, but I wanted to include a screenshot of the example output for ease-of-access.

A screen shot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screen shot of a computer screen

Description automatically generated

**GitHub Repository for Assignment 6:**

References

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to Algorithms, fourth edition*. MIT Press.