CSE 3500 – Algorithms and Complexity
Fall 2023
Homework 7

Implementing and Analyzing Quicksort Variants

Objective: In this assignment, you will implement two versions of the Quicksort algorithm with different pivot selection methods and measure their runtime performance on sorted and random lists of varying sizes. You will then analyze the results to understand the impact of the selected pivot on the algorithm's efficiency.

Instructions:

Implementation:

Implement Quicksort using the following two pivot selection methods:

a. Select the last element as the pivot.

b. Select the pivot as the median of medians (Median of Medians algorithm). You may refer to the implementation provided earlier.

Write code to generate random lists and sorted lists of sizes **500, 800, 1000, 1200, and 1500**. You can use Python's `random` module to generate random lists and sorting functions for sorted lists.

Runtime Measurement:

Measure the runtime for each of the two Quicksort implementations on each list type (sorted and random) for the specified sizes.

You can use Python's `time` module to record the execution time. Make sure to average the runtime over multiple runs for accuracy.

Data Reporting:

Create a table to report your results, including the size of the list, pivot selection method, average runtime, and any other relevant information.

Example:

Algorithm	**Input**				
	500	**800**	**1000**	**1200**	**1500**
Sorted-Mid_mids					
Sorted-Last					
Random-Mid_mids					
Random-Last					

Conclusion:

Write a conclusion that discusses how the choice of pivot affects the runtime of the Quicksort algorithm.

Analyze and compare the performance of the two pivot selection methods for different input scenarios (sorted and random lists) and list sizes.

Discuss any trends, observations, or insights from your experiments.

Submission:

Submit a report containing the following.

(**40 points**) Provide the table comparing the two versions by listing the running time for each data size.

(**30 point**) Draw conclusion on why choice of pivot matters.

(**30 points**) Attach the source code of your implementation. If it is short enough, you may simply include your code as part of the report.

Note: Python uses a maximum recursion depth of 1000 to ensure no stack overflow errors and infinite recursions are possible. You can change the maximum recursion depth in Python.

To do this, call the `sys.setrecursionlimit()` function.

For example, let's set the maximum recursion depth to 10000:

```
import sys
print(sys.setrecursionlimit(10000))
```