Quick Sort Tracing – Handwritten Submission

Project Overview

This project involves manually tracing the Quick Sort algorithm—a divide-and-conquer method that uses partitioning. You will document the process of selecting a pivot, partitioning the array, and recursively sorting the partitions. Additionally, count the number of comparisons and swaps made during the partitioning. Submit clear photos of your handwritten work along with a table summarizing your counts and observations.

Objectives

- Understand the partitioning process and recursive nature of Quick Sort.
- Trace the algorithm's pivot selection, partitioning, and recursive calls step by step.
- Accurately record the number of comparisons and swaps during partitioning.
- Enhance your documentation skills through handwritten tracing.

Instructions

1. **Algorithm Review:**
    - Review Quick Sort: select a pivot, partition the array into elements less than and greater than the pivot, and recursively sort the partitions.
    - Decide on your pivot selection strategy (e.g., first element, last element, median) and mention it in your tracing.
2. **Inputs for Tracing:**
   Trace Quick Sort on the following inputs:
    - **Input 01:** Your Complete Name
      (Sort the characters in alphabetical order.)
    - **Input 02:** `2 2 2 2 2 2 2 2 2 2 2`
      (Identical elements; note how partitioning behaves.)
    - **Input 03:** `1 2 3 4 5 6 7 8 9 10`
      (An already sorted list—observe the partitioning process.)
    - **Input 04:** `10 9 8 7 6 5 4 3 2 1`
      (A reverse sorted list—analyze how pivot selection impacts performance.)
    - **Input 05:** `1 2 3 4 5 10 9 8 7 6 5`
      (A mixed order input for an intermediate scenario.)
    - **Input 06:** Any additional input of your choice

(Explain why you selected this input.)

3. **Tracing the Algorithm:**
   For each input:

   - Write the initial array.
   - Detail the partitioning process:
     - Indicate which element is chosen as the pivot.
     - Show how the array is partitioned by comparing elements to the pivot.
     - Record every swap performed to place elements in the correct partition.
   - Trace the recursive calls until the entire array is sorted.
   - Annotate each step with clear explanations.

4. **Recording Comparisons and Swaps:**
   For each trace, count and record:

   - **Number of Comparisons:** Total comparisons made during partitioning.
   - **Number of Swaps:** Total swaps executed to reposition elements around the pivot.
   - Mention any optimizations if applied.

5. **Documentation Table:**
   Fill in the table below with your counts and attach the corresponding tracing images:

Input	Input Description	Number of Comparisons	Number of Swaps	Tracing Image & Observations
Input 01	Your Complete Name (alphabetically sorted characters)	*[Your Count]*	*[Your Count]*	*Attach image. Note observations about pivot selection and partitioning for characters.*
Input 02	`2 2 2 2 2 2 2 2 2 2 2` (identical elements)	*[Your Count]*	*[Your Count]*	*Attach image. Observe how identical elements affect comparisons and swaps.*
Input 03	`1 2 3 4 5 6 7 8 9 10` (already sorted)	*[Your Count]*	*[Your Count]*	*Attach image. Note the behavior in a best-case scenario during partitioning.*

Input	Input Description	Number of Comparisons	Number of Swaps	Tracing Image & Observations
Input 04	`10 9 8 7 6 5 4 3 2 1` (reverse sorted)	*[Your Count]*	*[Your Count]*	*Attach image. Explain how pivot selection influences the number of comparisons and swaps.*
Input 05	`1 2 3 4 5 10 9 8 7 6 5` (mixed order)	*[Your Count]*	*[Your Count]*	*Attach image. Provide observations on partitioning in a mixed input scenario.*
Input 06	Custom input	*[Your Count]*	*[Your Count]*	*Attach image. Describe your choice and any unique observations during the process.*

6. **Presentation Requirements:**
    - Ensure neat, legible handwriting throughout your tracing.
    - Take clear, well-lit photos of your work and table.
    - Submit your photos along with the completed table and a brief summary of your reflections on Quick Sort.
7. **Evaluation Criteria:**
    - **Completeness:** Every input must be traced, and the table fully filled.
    - **Clarity:** Each partitioning and recursive step is clearly documented and annotated.
    - **Accuracy:** The Quick Sort algorithm is correctly implemented and traced.
    - **Analysis:** Accurate counts of comparisons and swaps with thoughtful observations.
    - **Presentation:** Neat, legible handwritten work and clear photos.