CS 5720 Design and Analysis of Algorithms
Homework #5

Your Name

October 9, 2024

Question 1: MergeSort and QuickSort Analysis

For each part (a)-(d), we will provide a recurrence relation for the number of element comparisons required by the algorithm under the given conditions and solve the recurrence relation using the Master Theorem where applicable.

(a) MergeSort on Sorted Arrays

1. **Recurrence Relation**:

$$T(n) = 2T\left(\frac{n}{2}\right) + n - 1$$

- Splitting the array takes $2T(n/2)$ comparisons. - Merging two sorted arrays of size $n/2$ takes $n-1$ comparisons (since each comparison merges one element).
2. **Solution Using Master Theorem**: - Here, $a = 2$, $b = 2$, and $f(n) = n - 1 = \Theta(n)$. - Calculate $\log_b a = \log_2 2 = 1$. - Since $f(n) = \Theta(n)$, we use Case 2 of the Master Theorem.

$$T(n) = \Theta(n \log n)$$

(b) MergeSort on Reverse-Sorted Arrays

1. **Recurrence Relation**:

$$T(n) = 2T\left(\frac{n}{2}\right) + n - 1$$

- The recurrence relation is the same as for sorted arrays because MergeSort's comparison count does not depend on the order of the elements.
2. **Solution Using Master Theorem**: - Same as part (a).

$$T(n) = \Theta(n \log n)$$

(c) QuickSort on Sorted Arrays

1. **Recurrence Relation**:

$$T(n) = T(1) + T(n-1) + (n-1)$$

- The first partition step compares each element to the pivot, making $n-1$ comparisons.
- The best pivot selection (first element in sorted array) leads to one partition of size $n-1$ and one partition of size 0.
2. **Solution**: - The recurrence simplifies to:

$$T(n) = T(n-1) + n - 1$$

- Unroll the recurrence:

$$T(n) = T(n-1) + (n-1) = T(n-2) + (n-2) + (n-1) = \ldots = T(1) + \sum_{i=1}^{n-1} i = T(1) + \frac{n(n-1)}{2}$$

- Since $T(1) = 0$ (base case), we have:

$$T(n) = \Theta(n^2)$$

(d) QuickSort on Arrays of All Equal Elements

1. **Recurrence Relation**:

$$T(n) = 2T(n/2) + (n-1)$$

- Each partition step still compares each element to the pivot, making $n-1$ comparisons. - For arrays with all equal elements, each partition splits the array into two equal parts.
2. **Solution Using Master Theorem**: - Here, $a = 2$, $b = 2$, and $f(n) = n - 1 = \Theta(n)$. - Calculate $\log_b a = \log_2 2 = 1$. - Since $f(n) = \Theta(n)$, we use Case 2 of the Master Theorem.

$$T(n) = \Theta(n \log n)$$

Question 2: Optimizing QuickSort for Sorted Arrays

To optimize QuickSort for sorted arrays, we should select the pivot such that the partitions are balanced.

Pivot Selection Rule

- **Rule**: Select the median of the array as the pivot. In sorted arrays, this will divide the array into two equal parts.

Resulting Recurrence Relation

1. **Recurrence Relation**:

$$T(n) = 2T\left(\frac{n}{2}\right) + n - 1$$

- Selecting the median as the pivot results in two partitions of size $n/2$.
2. **Solution Using Master Theorem**: - Here, $a = 2$, $b = 2$, and $f(n) = n - 1 = \Theta(n)$. - Calculate $\log_b a = \log_2 2 = 1$. - Since $f(n) = \Theta(n)$, we use Case 2 of the Master Theorem.

$$T(n) = \Theta(n \log n)$$