Solution Outline for Lab 6

Jerry

SUSTC

October 14, 2025

Content

Yet Another Quick Sort Problem

Yet Another Quick Sort Problem

Given the pseudocode for QUICKSORT and the **permutation** A, then determine the number of times the swap operation is executed when performing QUICKSORT on A.

$1 \leq n \leq 5 \times 10^5$.

Yet Another Quick Sort Problem (cont'd)

Hint 1

This problem cannot be directly simulated using pseudocode, as it would degrade to a time complexity of $\mathcal{O}(n^2)$, which would not work with the current large dataset.

Yet Another Quick Sort Problem (cont'd)

Hint 1

This problem cannot be directly simulated using pseudocode, as it would degrade to a time complexity of $\mathcal{O}(n^2)$, which would not work with the current large dataset.

Lemma 1

The number of swap operations executed by QUICKSORT *is* $\mathcal{O}(n \log n)$*, which holds for **any sequence**.*

Proof. Assume the pivot is less than or equal to the median, so it will be finally placed at position $m \leq (l+r)/2$. Let L be the elements in $(m, r]$ with indices in $[l, m]$, and R be those in $[l, m)$ with indices in $[m, r]$. QUICKSORT swaps elements between L and R, with $|L|, |R| \leq m$.

Yet Another Quick Sort Problem (cont'd)

Lemma 1 (cont'd)

Let $T(n)$ represent the number of swap operations required for a sequence of length n. The recurrence relation can be expressed as:

$$
\begin{aligned}
T(n) &= T(m) + T(n-m) + \min(m, n-m) \\
&= \sum_{i=1}^{k} T(m_i) + \mathcal{O}\left(\sum_{i=1}^{k} m_i\right) \\
&\quad \text{(take the } \leq n/2 \text{ part at each split as } m_i\text{)} \\
&= \sum_{i=1}^{k} T(m_i) + \mathcal{O}(n) = \mathcal{O}(n \log n)
\end{aligned}
$$

The last step holds because $m_i \leq n/2$, so the recursion depth is at most $\mathcal{O}(\log n)$. $\square$

Yet Another Quick Sort Problem (cont'd)

Hint 2

According to Lemma 1, we can directly simulate each swap. The bottleneck in the pseudocode is finding elements greater or less than the pivot, which needs optimization.

Lemma 1 provides the solution: find lists L and R. The former can be scanned directly, while the latter can be tracked using a table with element positions. To swap in pseudocode order, we sort one of list and swap elements one by one. The time complexity is $\mathcal{O}(n \log^2 n)$.

Yet Another Quick Sort Problem (cont'd)

Bonus Problem (no extra points)

1. When m is close to the median, directly simulating the pseudocode results in better complexity, the above algorithm can be optimized to $\mathcal{O}\left(\frac{n \log^2 n}{\log \log n}\right)$.
2. The problem can also be solved when A is not a permutation, but it requires advanced data structures.

THANK YOU!