

Algorithm Design and Analysis

Assignment 1

假设 $n = 4^x$

1. (20 points) Solve the following recurrence relations and give a Θ bound for each of them.

(a) (4 points) $T(n) = 5T(n/4) + n$

$$T(n) = 5T(\frac{n}{4}) + n$$

(b) (4 points) $T(n) = 7T(n/7) + n$

$$T(n) = 5T(\frac{n}{4}) + n$$

(c) (6 points) $T(n) = 49T(n/25) + n^{3/2} \log n$

$$T(\frac{n}{4}) = 5T(\frac{n}{16}) + \frac{n}{4}$$

(d) (6 points) $T(n) = 3T(n-1) + 2$

2. (20 points) A k -way merge operation. Suppose you have k sorted arrays, each with n elements, and you want to combine them into a single sorted array of kn elements. Design an efficient algorithm using divide-and-conquer (and give the time complexity).

3. (20 points) You are interested in analyzing some hard-to-obtain data from two separate databases. Each database contains n numerical values-so there are $2n$ values total-and you may assume that no two values are the same. You'd like to determine the median of this set of $2n$ values, which we will define here to be the n -th smallest value. However, the only way you can access these values is through queries to the databases. In a single query, you can specify a value k to one of the two databases, and the chosen database will return the k -th smallest value that it contains. Since queries are expensive, you would like to compute the median using as few queries as possible. Give an algorithm that find the median value using $O(\log n)$ queries.

4. (20 points) Show that the QuickSort algorithm runs in $O(n^c)$ time on average for some constant $c < 2$ if the pivot is chosen randomly.

Remark: The algorithm actually runs in $O(n \log n)$ time on average, but for this question it suffices to give an analysis better than $O(n^2)$.

5. (20 + 5 points) Given an $n \times m$ 2-dimensional integer array $A[0, \dots, n-1; 0, \dots, m-1]$ where $A[i, j]$ denotes the cell at the i -th row and the j -th column, a local minimum is a cell $A[i, j]$ such that $A[i, j]$ is smaller than each of its four adjacent cells $A[i-1, j]$, $A[i+1, j]$, $A[i, j-1]$, $A[i, j+1]$. Notice that $A[i, j]$ only has three adjacent cells if it is on the boundary, and it only has two adjacent cells if it is at the corner. Assume all the cells have distinct values. Your objective is to find *one* local minimum (i.e., you do not have to find all of them).
- (a) (10 points) Suppose $m = 1$ so A is a 1-dimensional array. Design a divide-and-conquer-based algorithm for the problem above. Write a recurrence relation of the algorithm, and analyze its running time.
 - (b) (10 points) Suppose $m = n$. Design a divide-and-conquer-based algorithm for the problem above. Write a recurrence relation of the algorithm, and analyze its running time.
 - (c) (Bonus: 5 points) Generalize your algorithm such that it works for general m and n . The running time of your algorithm should *smoothly* interpolate between the running times for the first two parts.

Tips: You can first think whether the local minimum must appear, and try to find out the reason.

6. How long does it take you to finish the assignment (include thinking and discussing)? Give a score (1,2,3,4,5) to the difficulty. Do you have any collaborators? Write down their names here.