DAA Problem Set 2

Pradeesha Ashok

January 2025

1 Solving Recurrences

For each of the following recurrences, give an expression for the runtime $T(n)$. (Try the following problems in more than one way)

1. $T(n) = 3T(n/2) + n^2$
2. $T(n) = 16T(n/4) + n!$
3. $T(n) = 6T(n/3) + n^2 \log n$
4. $T(n) = \sqrt{n}T(\sqrt{n}) + n$
5. $T(n) = n^{1/4}T(n^{3/4}) + 1$
6. $T(n) = T(\sqrt{n}) + n^5$
7. $T(n) = T(n-1) + n,\ T(1) = 1$
8. $T(n) = 3T\left(\frac{n}{2} + 47\right) + 2n^2 + 10n - \frac{1}{2}$
9. $T(n) = 2T\left(\frac{n}{2}\right) + n \log n$
10. $T(n) = 16T(\sqrt{n}) + (\log_4 n)!$
11. $T(n) = 3T(n/4) + nlogn$
12. $T(n) = 4T(n/2) + n^2logn$
13. $T(n) = 2T(n/4) + \sqrt{n}$

2 Divide and Conquer

1. (a) Suppose you have 4 sorted arrays, each with n elements. Give an algorithm to merge all the arrays into a single sorted array.
   (b) Now instead of 4 arrays, we have k sorted arrays, each with n elements (k is now a part of the input). Give an algorithm to merge all the arrays to form a sorted array of kn elements, and comment on its time complexity.

2. Given an array of n elements, give a **divide and conquer** based algorithm with time complexity $O(nlogn)$ to check whether there exists an element that repeats more than $\frac{n}{2}$ times in the array. The elements in the array are not comparable, therefore a query like $A[i] > A[j]$ is meaningless. However, you can query whether two elements are equal.

3. Given an array A of n elements, (i, j), $i, j \leq n$ is called an inversion if $i < j$ and $A[i] > A[j]$. Design a divide and conquer algorithm that counts the number of inversions in A.

4. Given two sorted arrays A and B of n integers, give an algorithm to find the median element of A U B.

5. Given an integer array `nums`, return the number of reverse pairs in the array. A reverse pair is defined as a pair (i, j) where:

$$0 \leq i < j < \texttt{nums.length}$$
$$\texttt{nums}[i] > 2 \cdot \texttt{nums}[j].$$

6. Given an array A with n distinct numbers, a modified MergeSort algorithm works as follows : A_1 is the subarray $A[1 \ldots \lceil \frac{n}{3} \rceil]$, A_2 is the subarray $A[\lceil \frac{n}{4} \rceil \ldots \lceil \frac{n}{2} \rceil]$ and A_3 is the subarray $A[\lceil \frac{n}{2} \rceil \ldots n]$. Thus A_1, A_2, A_3 have respectively $\frac{n}{3}, \frac{n}{4}, \frac{n}{2}$ elements. Recursively sort A_1, A_2, A_3. Merge the sorted subarrays , remove duplicates and return the final array. How does the aymptotic running time of this modified MergeSort compare with that of the MergeSort algorithm discussed in class? Justify.

7. An array $A[1 \cdots n]$ is said to be **unimodal** if there exists an index p, $1 \leq p \leq n$ such that:

   - The elements of the array increase from index 1 to p, and
   - The elements decrease thereafter.

   For example, $A = \{5, 6, 7, 9, 8, 4, 3, 2\}$ is a unimodal array with 9 being the peak element.

   Problem: Give an $O(\log n)$ algorithm that finds the peak element in a unimodal array.

8. You are given a sorted array of numbers where every value except one appears exactly twice; the remaining value appears only once. Design an efficient algorithm for finding which value appears only once.

9. Given below is the pseudocode for Strassen's algorithm for matrix multiplication. Prove the correctness of the algorithm, and solve for the time complexity.

Algorithm 1 Strassen's Algorithm for Matrix Multiplication

Require: Matrices A and B, both of size $n \times n$
Ensure: Matrix $C = A \times B$

if $n = 1$ **then**
    $C[1][1] \leftarrow A[1][1] \times B[1][1]$
    return C
end if
Divide A into submatrices $A_{11}, A_{12}, A_{21}, A_{22}$ of size $n/2 \times n/2$
Divide B into submatrices $B_{11}, B_{12}, B_{21}, B_{22}$ of size $n/2 \times n/2$
Compute the following intermediate matrices:
$M_1 \leftarrow (A_{11} + A_{22})(B_{11} + B_{22})$
$M_2 \leftarrow (A_{21} + A_{22})B_{11}$
$M_3 \leftarrow A_{11}(B_{12} - B_{22})$
$M_4 \leftarrow A_{22}(B_{21} - B_{11})$
$M_5 \leftarrow (A_{11} + A_{12})B_{22}$
$M_6 \leftarrow (A_{21} - A_{11})(B_{11} + B_{12})$
$M_7 \leftarrow (A_{12} - A_{22})(B_{21} + B_{22})$
Compute the resulting submatrices of C:
$C_{11} \leftarrow M_1 + M_4 - M_5 + M_7$
$C_{12} \leftarrow M_3 + M_5$
$C_{21} \leftarrow M_2 + M_4$
$C_{22} \leftarrow M_1 + M_3 - M_2 + M_6$
Combine $C_{11}, C_{12}, C_{21}, C_{22}$ into the final matrix C
return C