1. Solve the following recurrences.

   (a) \( T(n) = 2T([n/2]) + n \).

   (b) \( T(n) = T([n/2]) + O(n) \) by expanding out the recurrence.

   (c) \( T(n) = T(n-a) + T(a) + n \), where \( a \geq 1 \) is a constant.

2. The running time of an algorithm \( A \) is described by the recurrence \( T(n) = 4T(n/2) + n^2 \). A competing algorithm \( A' \) has a running time of \( T'(n) = aT'(n/4) + n^2 \). What is the largest integer value for \( a \) such that \( A' \) is asymptotically faster than \( A \)?

3. (The skyline problem) A city’s skyline is the outer contour of the silhouette formed by all the buildings in that city when viewed from a distance. Now suppose you are given the locations and height of all the buildings. Develop a recursive algorithm that outputs the skyline formed by these buildings collectively.

4. (Majority element) Develop an \( O(n \log n) \)-time recursive algorithm finds the majority element in an array of size \( n \). The majority element is the element that appears more than \( \lfloor n/2 \rfloor \) times. (You can assume that such an element exists.) Your algorithm should not use sorting.

   Harder problem: Find an \( O(n) \)-time algorithm.

5. (Pattern matching with wildcards, exam 2019) In the problem of pattern matching with wildcards, you are given a text \( T \) of length \( n \) and a pattern of length \( m \) over an alphabet of integers. Both \( T \) and \( P \) may contain wildcards “?”, special characters that match any other character of the alphabet. Your task is to develop an efficient algorithm to find all positions where \( P \) matches \( T \).

Example: There are four positions where a pattern \( P = 2 ? 2 \) matches the text \( T = 21222?2 \): they are 3, 5, 6, 7.

   (a) Given two integer vectors \( X, Y \) of lengths \( n \) and \( m, n \geq m \), their convolution is defined as a vector \( C \) of length \( n-m \), where \( C[i] = \sum_{j=1}^{m} X[i+m-j]Y[j] \). Show an \( O(n \log m) \)-time algorithm for computing the convolution.

   Hint: Consider two polynomials, \( \sum_{i=1}^{n} X[i]z^i \) and \( \sum_{j=1}^{m} Y[j]z^j \). What are the coefficients of their product?

   (b) Assume that \( P \) and \( T \) contain only positive integers and wildcards. Let \( T'[i] = T[i] \) if \( T[i] \) is an integer and \( T'[i] = 0 \) if \( T'[i] = ? \). Define \( P' \) analogously. Show that there is an occurrence of \( P \) that ends at the position \( i \) of \( T \) if and only if \( \sum_{j=1}^{m} P'[j]T'[i-m+j](P'[j] - T'[i-m+j])^2 \) = 0.

   (c) Show that the values \( \sum_{j=1}^{m} P'[j]T'[i-m+j](P'[j] - T'[i-m+j])^2 \), \( j = m, \ldots, n \) can be computed in \( O(n \log m) \) time. Hence, show an \( O(n \log m) \) time algorithm for pattern matching with wildcards.

   Hint: Expand \( P'[j]T'[i-m+j](P'[j] - T'[i-m+j])^2 \) and then apply your algorithm for computing the convolution two times.
Extra questions

6. Consider the regularity condition \( af(n/b) \leq cf(n) \) for some constant \( c < 1 \), which is part of case 3 of Master theorem. Give an example of a simple function \( f(n) \) that satisfies all the conditions in case 3 of Master theorem except the regularity condition.

7. An array \( A[1..n] \) contains all the integers from 0 to \( n \) except one. It would be easy to determine the missing integer in \( O(n) \) time by using an auxiliary array \( B[0..n] \) to record which numbers appear in \( A \). In this problem, however, we cannot access an entire integer in \( A \) with a single operation. The elements of \( A \) are represented in binary, and the only operation we can use to access them is “fetch the \( j^{th} \) bit of \( A[i] \)”, which takes constant time. Show that if we use only this operation, we can still determine the missing integer in \( O(n) \) time.

8. Hamming distance between two strings of equal length is defined as the number of mismatches between them. For example, the Hamming distance between “0010” and “1010” is one. Given a binary string \( T \) of length \( 2n \) (text) and a binary string of length \( n \) (pattern), develop an algorithm that computes the Hamming distance between each \( n \)-length substring of \( T \) and \( P \) in \( O(n \log n) \) time.

9. Write a recursive algorithm that searches for a value in an \( m \times n \) matrix. This matrix has the following properties: (a) Integers in each row are sorted in ascending from left to right; (b) Integers in each column are sorted in ascending from top to bottom.

10. The Tower of Hanoi puzzle was invented by the French mathematician Edouard Lucas in 1883. He was inspired by a legend that tells of a Hindu temple where the puzzle was presented to young priests. At the beginning of time, the priests were given three poles and a stack of \( n = 64 \) gold disks, each disk a little smaller than the one beneath it. Their assignment was to transfer all \( n \) disks from one of the three poles to another, with two important constraints. They could only move one disk at a time, and they could never place a larger disk on top of a smaller one. The priests worked very efficiently, day and night, moving one disk every second. When they finished their work, the legend said, the temple would crumble into dust and the world would vanish. Develop a recursive algorithm for a general value of \( n \). Give asymptotic upper and lower bounds for its running time. Make your bounds as tight as possible, and justify your answers.