

Problem 1. Find the upper and lower bounds for $\sum_{i=0}^n (i^4 + 3i^2)$ using the integral approximation method. Indicate whether this is a monotonically increasing or decreasing function or both. Show your work.

Problem 2. Write a recurrence equation for a multiplication algorithm that squares any n -digit number by dividing the n -digit number into three parts, each comprised of $n/3$ -digits. This way you are reducing the operation to multiplying six $n/3$ -digit numbers. You may assume n to be “nice”. Also, the algorithm does not know multiplications except squaring, so each of your multiplications should involve squaring only, even when the original numbers may not be the same. In other words, this algorithm should be able to multiply any two numbers by making use of squaring alone.

Solve the recurrence equation using the recursion tree approach to find the exact number of multiplications and additions to find the square of a number. You may represent an atomic multiplication between two, one-digit numbers, as μ and the atomic addition of two, one-digit numbers, as α .

Problem 3. Now that you have worked with divide and conquer algorithms and recurrences, we will try to combine it all together. In a divide and conquer algorithm, the problem is divided into smaller subproblems, each subproblem is solved *recursively*, and a *combine* algorithm is used to solve the original problem. Assume that there are a subproblems, each of size $1/b$ of the original problem, and that the algorithm used to combine the solutions of the subproblems runs in time cn^k , for some constants a, b, c , and k . For simplicity, we will assume, $n = b^m$, so that n/b is always an integer (b is an integer greater than 1). Answer the following:

- (a) Write the generalized recurrence equation.
- (b) Solve the recurrence equation using a recursion tree approach. Base case, $T(1) = c$.
- (c) Once you obtain the solution to the recurrence equation in part(b), you will need to evaluate runtimes exactly for three cases:
 - (a) $a > b^k$
 - (b) $a = b^k$
 - (c) $a < b^k$

In order to help you verify your exact runtime for these three cases, the asymptotic runtimes for the three cases are given as:

$$T(n) = \begin{cases} O(n^{\lg_b a}) & \text{if } a > b^k \\ O(n^k \lg_b n) & \text{if } a = b^k \\ O(n^k) & \text{if } a < b^k \end{cases}$$

Show your work. Make appropriate assumptions.

Congratulations! you have verified a very important theorem.

Problem 4. You are given a sequence of numbers, one by one; assume for simplicity that they are distinct. Each time you receive a new number, your responsibility is to respond with the median element of all the numbers you have seen thus far. Thus, after seeing the first 11 numbers, you should reply with the sixth-smallest one you have seen; after 12, the sixth- or the seventh-smallest; after 13, the seventh-smallest; and so on.

Design a $O(\lg i)$ algorithm, where i is the number of elements in round i , to find a median. You may use pseudo-code or English to explain how your algorithm would compute the median in logarithmic time per round. Be clear and concise. Don't write long sentences if you explain in English.