1. Runtime complexity of algorithm mathematically.

Induru Revanth Kumar Reddy

To find the runtime of the given algorithm mathematically, we can analyze the nested loops. The outer loop runs from 1 to n, and the inner loop also runs from 1 to n. Inside the inner loop, there is a constant amount of work.

The outer loop runs for n iterations.
The inner loop runs for n iterations for each iteration of the outer loop.

So, the total number of iterations is given by the product of the iterations of both loops:

Total iterations = n × n

Inside the innermost part, there is a constant amount of work (x=x+1), which takes constant time, denoted as O(1).

Now, we can express the total running time mathematically using summations:

$$T(n) = \Sigma_{i=1}^{n} \Sigma_{j=1}^{n} O(1)$$

Simplifying this, we get:

$$T(n) = O(1) \times \Sigma_{i=1}^{n} \Sigma_{j=1}^{n} 1$$

$$T(n) = O(1) \times \Sigma_{i=1}^{n} n$$

$$T(n) = O(1) \times n \times n$$

Simplifying further, we get:

$$T(n) = O(1) \times n^2$$

Therefore, the runtime of the given algorithm is $O(n^2)$.