1. To determine the runtime of the given algorithm mathematically, we need to analyze the nested loops and express the total number of iterations in terms of the input size n.The outer loop runs n times, and for each iteration of the outer loop, the inner loop runs n times. Therefore, the total number of iterations is given by the product of the number of iterations of both loops.

The total number of iterations is the sum of n for each iteration of the outer loop, repeated n times for the inner loop:

$$\text{Total iterations} = \sum_{i=1}^{n} \sum_{j=1}^{n} 1$$

Simplifying this double summation:

$$\text{Total iterations} = \sum_{i=1}^{n} n = n \cdot \sum_{i=1}^{n} 1 = n \cdot n = n^2$$

Therefore, the runtime of the given algorithm is O(n^2), indicating a quadratic time complexity with respect to the input size n.

4. Will this increate how long it takes the algorithm to run (e.x. you are timing the function like in #2)?

Yes, this modification will likely increase the time it takes for the algorithm to run. The addition of the statement y = i + j; inside the inner loop introduces additional arithmetic operations, which contribute to the overall workload of the algorithm.

5. Will it effect your results from #1?

The modification will likely affect the results from #1. The runtime complexity of the original algorithm was O(n2) , and the additional arithmetic operations inside the inner loop could contribute to a change in the constant factors or even the leading term of the polynomial.