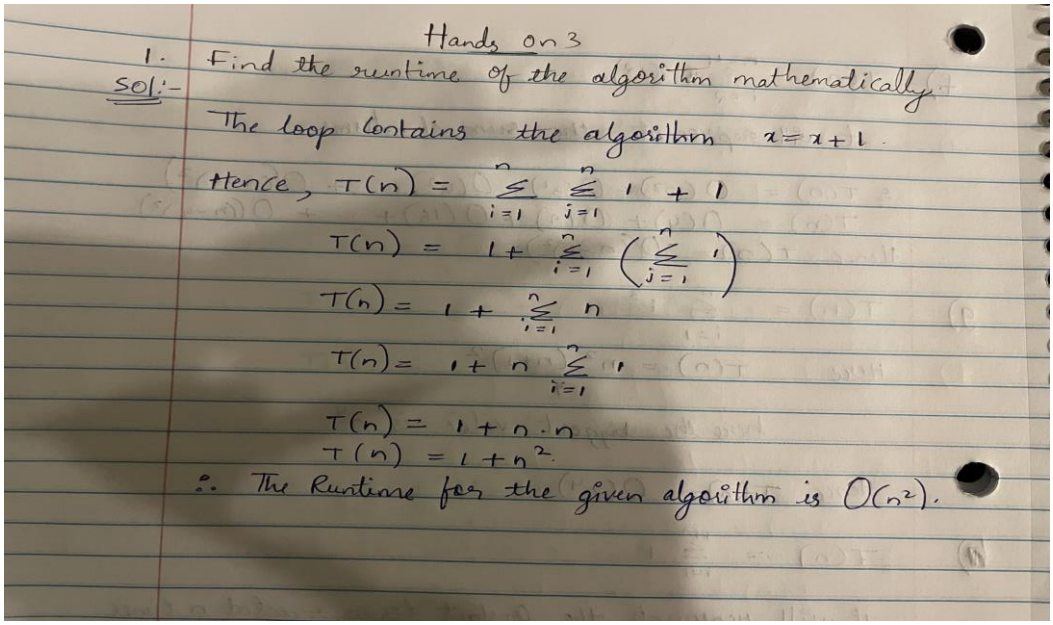
Sushrutha Reddy Mogili

1002164706

Hands on 3

1. Find the runtime of the algorithm mathematically (I should see summations).

Sol:



3.Find polynomials that are upper and lower bounds on your curve from #2. From this specify a big-O, a big-Omega, and what big-theta is.

Sol: Code in Daa-Hands-On 3

- **O(n^3)** as the upper bound.
- **Ω(n)** as the lower bound.
- Big-Theta represents both upper and lower bounds indicating In this case, Θ(n^2)

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

Sol: Yes, modifying first function tosecond function will increase the algorithm's execution duration. Within the nested loops of Second function , an extra operation is introduced: y=i+j, This indicates that an additional operation is usedfor every combination of i and j.

The only operation in the nested loops of the first function is the simple increment operation, x=x+1. For every combination of i and j inside the nested loops, we are now using the increment operation and the addition operation y=i+j in the new function.
The time complexity for the second function will be more than the first function, and When the input size increases, the algorithm's execution time will probably increase as well. The exact impact on performance depends on the size of n and the specific characteristics of the system running the algorithm.

5.Will it effect your results from #1?

Sol: Yes, the results will probably differ from first function if first function is changed to second function . The time complexity of both first function and second function is **O(n^2)**.