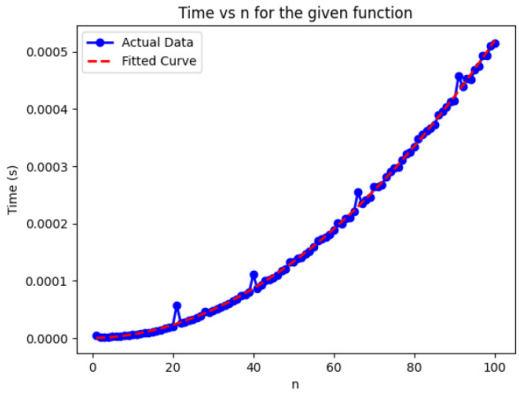
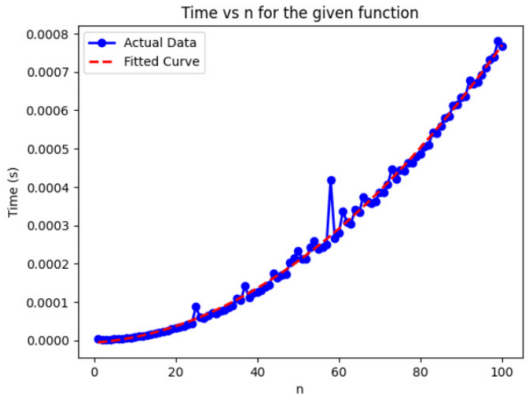
INDURU REVANTH KUMAR REDDY
1002193318

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

5. Will it effect your results from #1?





The T’ (n) function for the modified function is:

T’ (n) = n^2 * (t_x + t_y + t_i + t_j)

Whereas for old function T (n) = T’ (n) = n^2 * (t_x + t_i + t_j)

Therefore T’ (n) > T (n), (in the figure for n=100, T’ (n) ≈ 0.0008 and T (n) ≈ 0.0005)

Where:

- T’(n) is the time complexity of the modified function
- T’(n) is the time complexity of the old function
- n is the input size
- t_x is the time complexity of the operation **x = x + 1**
- t_y is the time complexity of the operation **y = i + j**
- t_i is the time complexity of the operation **i = i + 1**
- t_j is the time complexity of the operation **j = j + 1**

The time complexity of the modified function is still O (n^2) due to the nested loops, but the constant factor associated with the quadratic term will likely be larger, leading to an increase in the actual runtime.

When timing the function, you may observe a noticeable difference in execution time compared to the original function without the y = i + j operation.