Harshith kolli

1002189372

Hands on 3:

1:

x=1;→1

for i=1:n→ n+1

for j=1:n→n(n+1)

x=x+1;→ n^2

The runtime would be calculated with following summation formula:

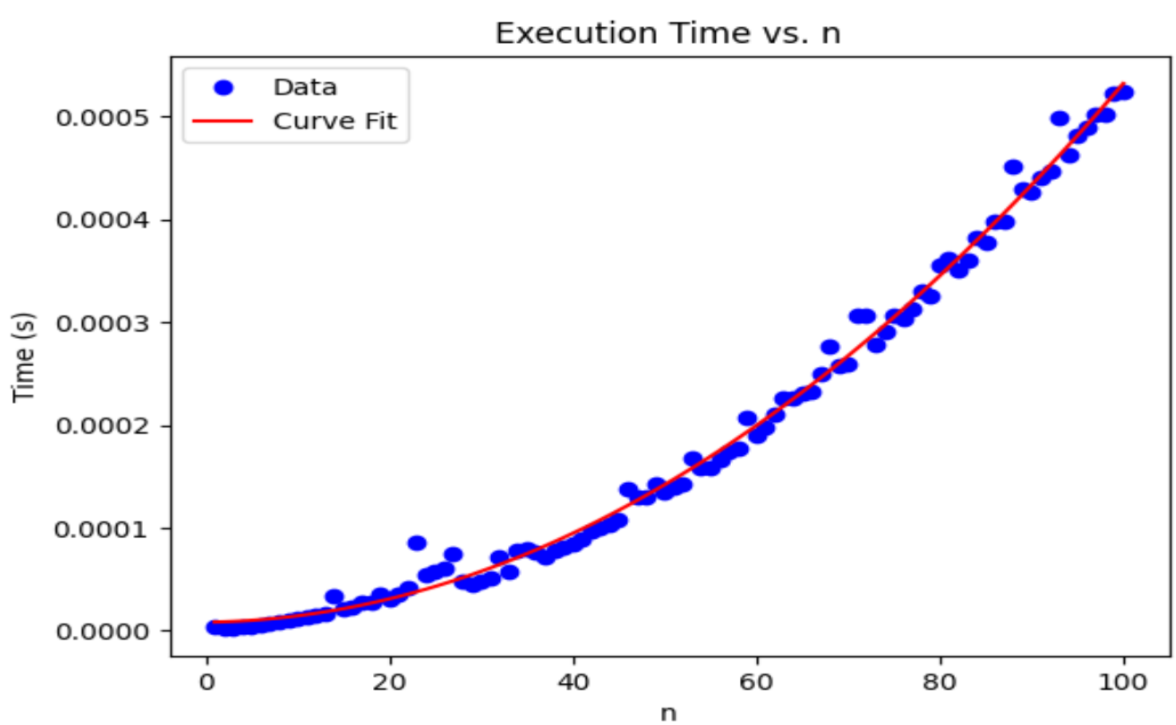
T(n)=1+ $\sum_{i=1}^{n+1} 1 \sum_{i=1}^{n} \sum_{j=1}^{n+1} 1 \sum_{i=1}^{n} \sum_{j=1}^{n} 1$

→t(n)=1+(n+1)+n(n+1)+n^2

→T(n)=2n^2+2n+2

O(n)=(n^2)[since the higher order term is(n^2) so the big O time complexity is(n^2)]

2:



3:

T(n)=an^2+bn+c

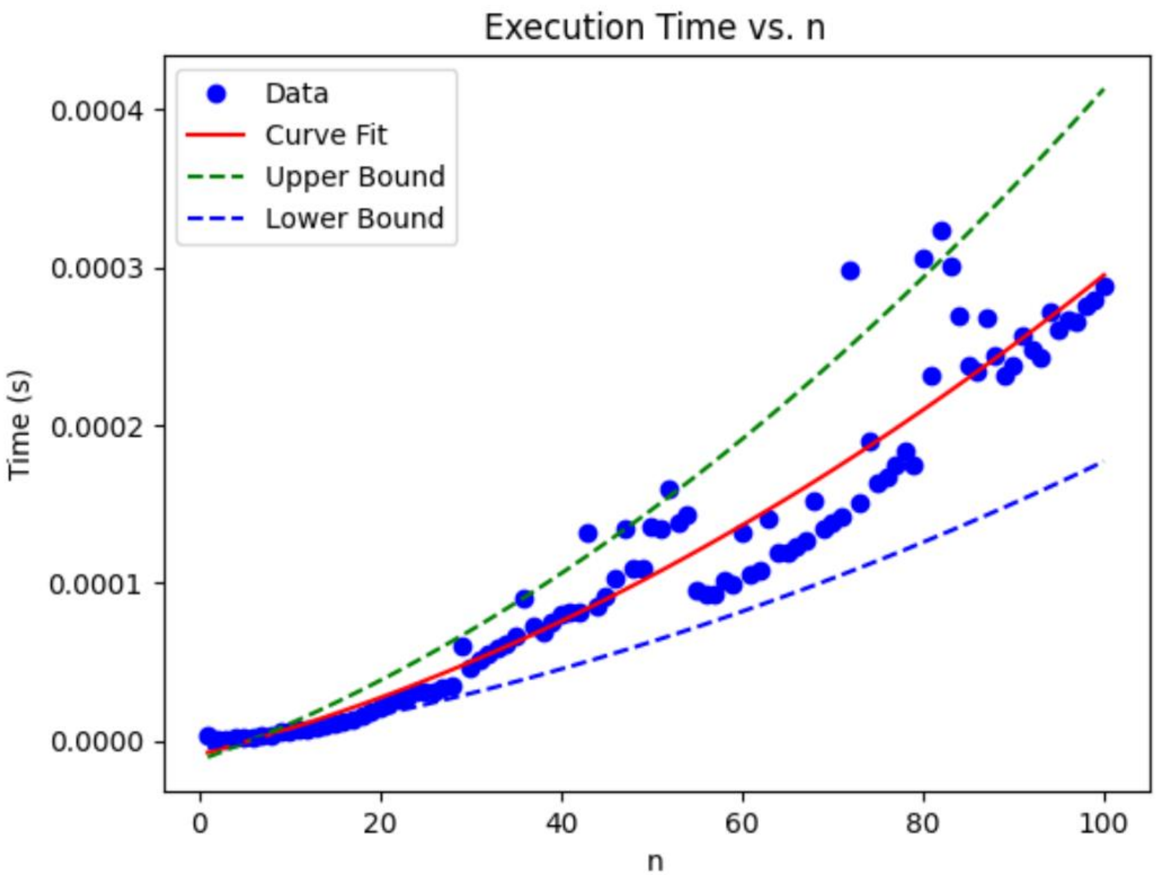
Upper bond:(n^2)

Lower bond:(n)

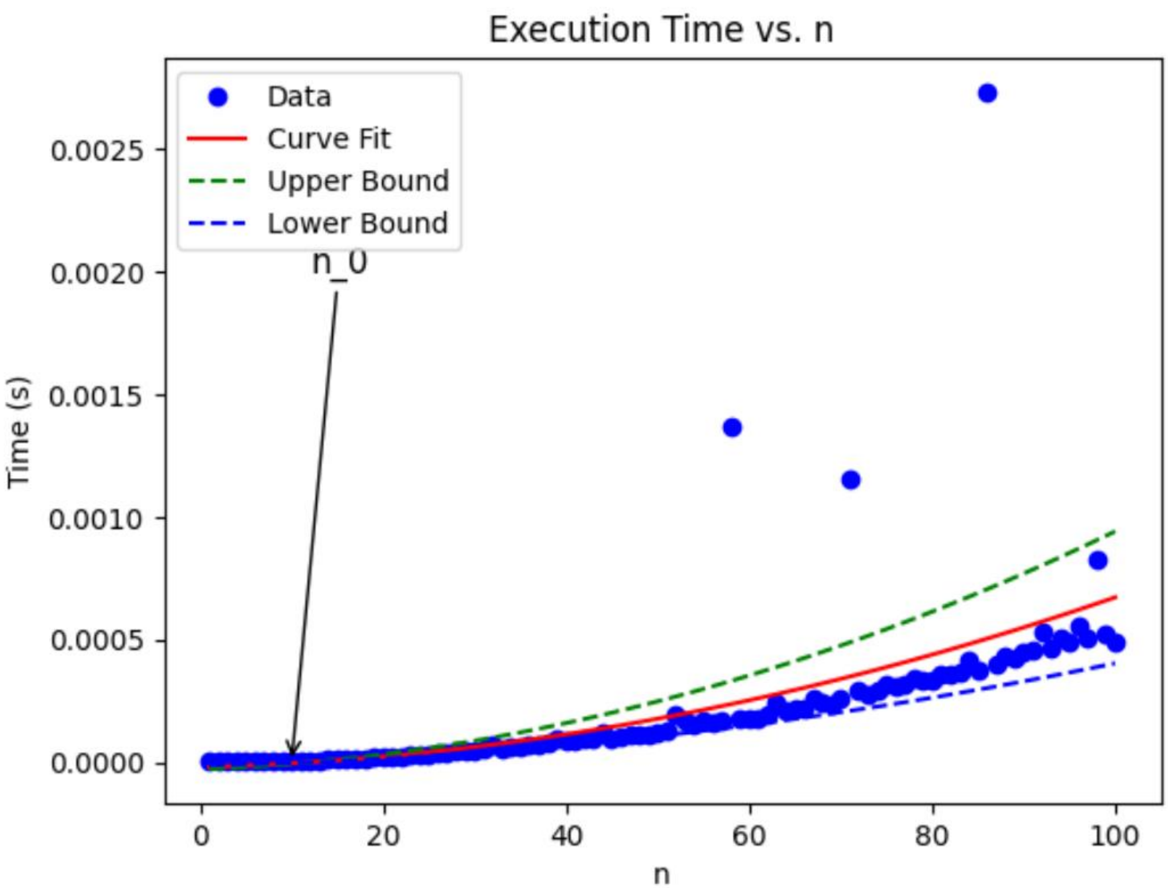
Big-O-(n^2)

Big-Theta-(n^2)

Big omega-(n^2)



4



4.By zooming the plot, at n=10 the upper bound is getting deviated from the fitted curve. So n_0=10 at which the algorithm's behavior becomes relevant or significant .

4: Yes, The runtime will be increased,The added statement y=i+j will double the runtime by adding(n^n) to the run time

i.e T(n)= 1+(n+1)+n(n+1)+(n^2)+(n^2)

5:

No,even the runtime increased their will be no change in the result as the equation stays quadratic.For Big-O notation we consider higher order terms and constants are ignored so it stays **O(n^2).**