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

Here,

⇨ Step 1: Understanding the Polynomial Curve Fit

From the previous step, we fit a quadratic polynomial to the measured runtime data:

$$T(n) \approx an2 + bn + c$$

where a,b,c are constants determined from the curve fitting.

⇨ Step 2: Defining Upper and Lower Bounds

To specify **Big-O (upper bound)** and **Big-Omega (lower bound)**, we need to find polynomials that bound our fitted quadratic function from above and below.

1. **Big-O (Upper Bound, O(g(n))**
   We need to find a function g(n) such that for some constant cand large enough n:

   $T(n) \leq c \cdot g(n)$

   Since we observed a quadratic trend, an upper bound polynomial is:

   $$O(n^2)$$

   This means that the worst-case runtime does not grow faster than some multiple of n^2 for sufficiently large n

2. **Big-Omega (Lower Bound, Ω(g(n))**
   We need to find a function g(n)g(n) such that for some constant c′ and large enough n:

   $$T(n) \geq c' \cdot g(n)$$

   The observed runtime grows at least as fast as n^2, so a lower bound polynomial is:

   $$\Omega(n^2)$$

This means that for large n, the function does not grow slower than some multiple $of\ n^2$.

3. **Big-Theta (Tight Bound, Θ(g(n))**
   Since both the upper and lower bounds are n^2 , we conclude:

$$\Theta(n^2)$$

   This means that the function T(n) is asymptotically bounded both above and below by n2n2, confirming that the algorithm has quadratic complexity.

Final Answer

- $\boldsymbol{Big-O}$: $O(n\hat{}2)$ $(Upper\ bound)$
- **Big-Omega**: Ω(n^2) (Lower bound)
- **Big-Theta**: Θ(n2) (Tight bound)

Since the function is both O(n2)O(n2) and Ω(n2)Ω(n2), the exact asymptotic complexity is:

T(n)=Θ(n2)T(n)=Θ(n2)
4o

1.