Big O, Omega, and Theta Notation

# 1. Big O Notation (O) – Upper Bound

Big O describes the worst-case scenario. It gives an upper limit on the time (or space) an algorithm might take, meaning the algorithm will not take more time or space than this in the worst case. It helps to understand the algorithm's behavior when the input size becomes very large.

Example:  
void printArray(int arr[], int n) {  
 for (int i = 0; i < n; i++) {  
 cout << arr[i] << " ";  
 }  
 cout << endl;  
}

In this example, the loop runs n times, so the time complexity is O(n) (linear). Even if the algorithm behaves better for smaller inputs, it will never perform worse than this in the worst case.

# 2. Omega Notation (Ω) – Lower Bound

Omega notation describes the best-case scenario. It gives a lower bound on the time (or space) an algorithm might take, meaning the algorithm will take at least this much time or space no matter what. This is useful for understanding the best possible performance of an algorithm.

Example:  
void findFirstElement(int arr[], int n) {  
 cout << arr[0] << endl;  
}

In this case, the time complexity is Ω(1) (constant time) because it only accesses the first element of the array. Regardless of the array size, the function will always take the same amount of time to complete.

# 3. Theta Notation (Θ) – Tight Bound

Theta notation describes a tight bound, meaning it represents both the best-case and worst-case scenarios. It provides an exact asymptotic behavior of the algorithm, stating that the algorithm will take this much time in both best and worst cases.

Example:  
void printArray(int arr[], int n) {  
 for (int i = 0; i < n; i++) {  
 cout << arr[i] << " ";  
 }  
 cout << endl;  
}

Here, the function runs exactly n times in any case (best or worst). So the time complexity is Θ(n) because the number of operations is proportional to n in both best and worst cases.

# Summary

- Big O (O): Describes the upper bound (worst case).  
- Omega (Ω): Describes the lower bound (best case).  
- Theta (Θ): Describes the exact bound (both best and worst cases are the same).

Example Comparison:  
For a sorting algorithm like insertion sort:  
- Best-case: Ω(n) (when the array is already sorted).  
- Worst-case: O(n²) (when the array is in reverse order).  
- Tight bound: Θ(n²) (average case, it requires quadratic time).