Step By Step Analysis Of Triple Nested Loop With Dependent Ranges.

```
for (int i = 1; i <= N; i++) {          // Outer loop

   for (int j = i; j <= N; j++) {      // Middle loop (depends on i)

      for (int k = j; k <= N; k++) { // Inner loop (depends on j)

         cout << "Hello";           // Constant work

      }
   }
}
```

Step-by-step Analysis

1. Outer loop (i = 1 … N)

- Runs **N times**.
- For each i, the middle loop runs.

2. Middle loop (j = i … N)

- For fixed i, j goes from i to N.
- So, number of iterations = (N - i + 1).

3. Inner loop (k = j … N)

- For fixed i and j, k goes from j to N.
- So, number of iterations = (N - j + 1).
- Each iteration executes cout << "Hello"; (constant time).

4. Total Work

$$T(N) = \sum_{i=1}^{N} \sum_{j=i}^{N} \sum_{k=j}^{N} 1$$

Simplification

Innermost sum (over k):

$$\sum_{k=j}^{N} 1 = (N - j + 1)$$

So,

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

Now simplify inner sum:

$$\sum_{j=i}^{N}(N-j+1)=\sum_{j=i}^{N}(N+1-j)$$

This is an arithmetic series:

- First term = N+1−i
- Last term = 1
- Number of terms = N−i+1

So,

$$\sum_{j=i}^{N}(N+1-j)=\frac{(N-i+1)(N+1-i+1)}{2}$$

$$=\frac{(N-i+1)(N-i+2)}{2}$$

Now outer sum:

$$T(N)=\sum_{i=1}^{N}\frac{(N-i+1)(N-i+2)}{2}$$

Let m=N−i+

- When i=1 ⇒ m=N
- When i=N ⇒ m=1

So,

$$T(N)=\sum_{m=1}^{N}\frac{m(m+1)}{2}$$

$$\sum_{m=1}^{N}m(m+1)=\sum_{m=1}^{N}(m^2+m)=\frac{N(N+1)(2N+1)}{6}+\frac{N(N+1)}{2}$$

Factor:

$$=\frac{N(N+1)(2N+1)}{6}+\frac{3N(N+1)}{6}=\frac{N(N+1)(2N+1+3)}{6}=\frac{N(N+1)(2N+4)}{6}=\frac{2N(N+1)(N+2)}{6}$$

$$=\frac{N(N+1)(N+2)}{3}$$

Now divide by 2 (because we had /2 earlier):

$$T(N)=\frac{1}{2}\cdot\frac{N(N+1)(N+2)}{3}=\frac{N(N+1)(N+2)}{6}$$

Final Result

$$T(N) = \frac{N(N+1)(N+2)}{6}$$