**TILDE APPROXIMATIONS**

1. N + 1 = ~N

2. 1 + 1/N = ~1

3. (1 + 1/N ) (1 + 2/N ) =~1

4. 2N^3 - 15 N^2 + N =~2N^3

5. lg(2N )/lg N =~1

6. lg(N^2 + 1) / lg N =~2

**ORDER OF GROWTH**

**Code-1:**

  int count = 0;

            for(int i = 0; i < N; i++)

{

                count++;

            }

answer :-N

**Code-2:**

int sum = 0;

    if(sum == 0)

{

                sum++;

}

answer :-1

**Code-3:**

for(int i = N; i > 0; i < N/2)

{

    int sum = 0;

}

answer :-

**Code-4:**

 for(int i = 0; i < N; i++)

{

       for(int j = 0; j < N; j++)

{

        System.out.println(“Hello”);

       }

    }

answer :-N^2

**Code-5:**

for(int i = 0; i < N; i++)

{

       for(int j = 0; j < N; j = j \* 2)

{

          System.out.println(“Hello”);

         }

       }

answer :-

**Code-6:**

public int fibonacci(int number)

{

if (number <= 1)

{

     return number;

     }

else

{

   return fibonacci(number - 1) + fibonacci(number - 2);

  }

}

answer :-

**Code-1:**

  int sum = 0;  
   for (int n = N; n > 0; n /= 2)

{  
      for(int i = 0; i < n; i++)  
    {

            Sum++;

}

}

Answer:-log(N)

**Code-2:**

int sum = 0;

        for (int i = 1;i < N; i \*= 2)  
 {

       for (int j = 0; j < i; j++)  
                {

sum++;

**}** }

Answer:-N\*log(N)

**Code-3:**

int sum = 0;  
        for (int i = 1 i < N; i \*= 2)  
        {

    for (int j = 0; j < N; j++)  
           {

    Sum++;

}

}

Answer:-    log(N)