**Mathematics Track**

1. **Finding Number Of Digits In A Number**

Given an integral number **N**. The task is to find the count of digits present in this number.

Let's say:

**N = 2019**

Number of digits in N here is 4 and,

the digits are: 2, 0, 1, 9.

**Some more Examples**:

**N = 2313**

Number of digits = 4

**N = 645**

Number of digits = 3

**N = 98346**

Number of digits = 5

### Solution 1

**Simple Solution**: A Simple Solution that comes in mind is:

1. Check if the number N is not equals to zero.
2. Increase the count of digits by 1 if N is not zero.
3. Reduce the number by dividing it by 10.
4. Repeat the above steps until the number is reduced to zero.

**Dry-run of above algorithm**: Consider an example, N = 58964. Initialize a variable **digitsCount** to zero which will store the count of digits. Keep incrementing *digitsCount* until N is not zero, and reduce it by dividing by 10 at each step.

**Iteration 1:** N **not equals** to 0

Increment digitsCount, digitsCount = digitsCount + 1.

digitsCount = 0 + 1 = 1.

N = N/10 = 58964/10 = 5896.

**Iteration 2:** N **not equals** to 0

Increment digitsCount, digitsCount = digitsCount + 1.

digitsCount = 1 + 1 = 2.

N = N/10 = 5896/10 = 589.

**Iteration 3:** N **not equals** to 0

Increment digitsCount, digitsCount = digitsCount + 1.

digitsCount = 2 + 1 = 3.

N = N/10 = 589/10 = 58.

**Iteration 4:** N **not equals** to 0

Increment digitsCount, digitsCount = digitsCount + 1.

digitsCount = 3 + 1 = 4.

N = N/10 = 58/10 = 5.

**Iteration 5:** N **not equals** to 0

Increment digitsCount, digitsCount = digitsCount + 1.

digitsCount = 4 + 1 = 5.

N = N/10 = 5/10 = 0.

**Iteration 6:** N becomes equal to 0.

Terminate any further operation.

Return value of digitsCount.

Therefore, the number of digits = 5.

**Analysis of above algorithm**: You can clearly see that, the number of operations performed in the above solution is equal to the count of digits present in the number. So, the time complexity of the solution is **O(digitsCount)**.

### Solution 2

**Better Solution**: A better solution is to use mathematics to solve this problem. The number of digits in a number say N can be easily obtained by using the formula:

number of digits in N = log10(N) + 1.

**Derivation**: Suppose the number of digits in the number **N** is **K**.

Therefore, we can say that:

10K-1 <= N < 10K

Applying base-10 logarithm to both sides in the above equation, we get:

K-1 <= log10(N) < K.

or, K - 1 + 1 <= log10(N) + 1 < K + 1

or, K <= log10(N) + 1 < K + 1

Therefore,

K = floor(log10(N) + 1)

**Analysis of above algorithm**: Since the above algorithm works in a single operation by using two mathematical operations, finding logarithmic and floor value. Therefore, the time complexity of the solution is **O(1)**.

**2. Arithmetic And Geometric Progressions**

A sequence of numbers is said to be in an Arithmetic progression if the difference between any two consecutive terms is always the same. In simple terms, it means that the next number in the series is calculated by adding a fixed number to the previous number in the series. For example, 2, 4, 6, 8, 10 is an AP because the difference between any two consecutive terms in the series (common difference) is the same (4 - 2 = 6 - 4 = 8 - 6 = 10 - 8 = 2).

**Formula Of Nth Term Of An A.P :**

If 'a' is the initial term and 'd' is the common difference.Thus, the explicit formula is:

**an = a + (n – 1) × d**

**Formula Of Sum Of First N Term Of A.P :**

**S = n/2 [2a + (n − 1) × d]**

General Formulas to solve problems related to Arithmetic Progressions: If ‘a’ is the first term and ‘d’ is the common difference:

* Arithmetic Mean = Sum of all terms in the AP / Number of terms in the AP.
* Sum of ‘n’ terms of an AP = 0.5 n (first term + last term) = 0.5 n [ 2a + (n-1) d ]

### Geometric Progression

A sequence of numbers is said to be in a **Geometric progression** if the ratio of any two consecutive terms is always the same. In simple terms, it means that next number in the series is calculated by multiplying a fixed number to the previous number in the series.For example, 2, 4, 8, 16 is a GP because ratio of any two consecutive terms in the series (common difference) is same (4 / 2 = 8 / 4 = 16 / 8 = 2).

**General Formulas to solve problems related to Geometric Progressions**:

If ‘a’ is the first term and ‘r’ is the common ratio:

* **nth term of a GP** = a\* rn-1
* **Geometric Mean** = nth root of product of n terms in the GP.
* **Sum of ‘n’ terms** of a GP (r < 1) = [a (1 – rn)] / [1 – r].
* **Sum of ‘n’ terms** of a GP (r > 1) = [a (rn – 1)] / [r – 1].
* **Sum of infinite terms** of a GP (r < 1) = (a) / (1 – r).

**3. Quadratic Equation**

A quadratic equation is a second-order polynomial equation of a variable say x. The general form of a quadratic equation is given as:

*ax*2 + *bx* + *c* = 0

Where a,b and c are real known values and, a can't be zero.

**Roots of an Equation**: The roots of an equation are the values for which the equation satisfies the given condition.

**A quadratic equation has two roots**. The roots of a quadratic equation can be easily obtained using the quadratic formula:



There arises **three cases** as described below while finding the roots of a quadratic equation:

* If ***b*2 < 4ac**, then roots are complex (not real).

For example roots of *x*2 + x + 1, roots are -0.5 + i1.73205 and -0.5 - i1.73205

* If ***b*2 = 4ac**, then roots are real and both roots are the same.

For example, roots of *x*2 - 2x + 1 are 1 and 1

* If ***b*2 > 4ac**, then roots are real and different.

For example, roots of *x*2 - 7x - 12 are 3 and 4

**Floor and Ceil**

Find it here [Ceil and Floor functions in C++](https://www.geeksforgeeks.org/ceil-floor-functions-cpp/)

**Practice Problems Link**

1. [**Count digits in a factorial**](https://practice.geeksforgeeks.org/problems/count-digits-in-a-factorial/0) **Easy**
2. [**Roots of Quadratic Equation**](https://practice.geeksforgeeks.org/problems/roots-of-quadratic-equation/0) **Easy**
3. [**Army Game**](https://www.hackerrank.com/challenges/game-with-cells/problem) **Medium**