Number of Digits of a Factorial

source: forthright48

Number of Digits of a numner

Logarithn of a number x, in base b, is a real number y such that $x = b^y$. $log_b(x) = y$ is same as, $x = b^y$, for example

$$log_{10}1234 = 3.0913151597$$

$$10^{3.0913151597} = 1234$$

$$\text{number of digits of : } x \text{ in base } 10 = \lfloor log_{10}(x) \rfloor + 1$$

$$\therefore \text{Number of digits of x in base B is} = \lfloor log_B(x) \rfloor + 1$$

```
#define eps 1e-9

int numberDigit ( int n ) {
    int wrongAnswer = log10(n) + 1; // This may give wrong answer sometimes.
    int rightAnswer = log10(n) + 1 + eps; // This is right.
    return rightAnswer;
}
```

we used $log10()$ function instead of $log()$ function. Unlike our calculators, in C++ $log()$ has base 2.

Digits of Factorial in base 10

formula used for finding digits of a factorial:

$$log_{10}(ab) = log_{10}(a) + log_{10}(b)$$

Let $x = log_{10}(N!)$ Then our answer will be $res = \lfloor log_{10}(N!) \rfloor + 1 = \lfloor x \rfloor + 1$ $x = log_{10}(N!)$ $x = log_{10}(1 \times 2 \times 3 \times ... \times N)$ $\therefore x = log_{10}1 + log_{10}2 + log_{10}3 + ... + log_{10}N$ So in order to calculate $x = log_{10}(N!)$, we can simply add value of all number from 1 to N with Time Complexity: $O(N)$

```
#define eps 1e-9

int factorialDigit ( int n ) {
    double x = 0;
    for ( int i = 1; i <= n; i++ ) {
        x += log10 ( i );
    }
    int res = x + 1 + eps;
    return res;
}
```

Digits of $N!$ in Different Base

$$\text{Number of digits of x in base B is} = \lfloor log_B(x) \rfloor + 1$$

Now, We can only use log with base 2 and 10 in C++. For that we will use the following formula.

$$log_B(x) = \frac{log_C(x)}{log_C(B)}$$

So in C++, we will use $C = 2$ or $C = 10$ to find value of $log_B(x)$.

```
#define eps 1e-9

int factorialDigitExtended ( int n, int base ) {
    double x = 0;
    for ( int i = 1; i <= n; i++ ) {
        x += log10 ( i ) / log10(base); // Base Conversion
    }
    int res = x + 1 + eps;
    return res;
}
```

Digits of factorial of a all the numbers from 1 to N [Pre-computation]

```
    for(int i=1;i<=1000000;i++) {
        ar[i] = ar[i-1]+log10(i/1.0);
    }

    int t;
    cin >> t;
    for (int test = 1; test <= t; ++test) {
        int n, base;
        cin >> n >> base;
        ll ans = 1+(ar[n] / (log10(base*1.0))) + eps;
        cout <<"Case " << test << ": " << ans << endl;
    }
```