All Tracks > Math > Number Theory > Totient Function

7 LIVE EVENTS

Math

Solve any problem to achieve a rank

View Leaderboard

Topics: Totient Function

# Totient Function

TUTORIAL PROBLEMS

## What is Euler's Totient Function?

Number theory is one of the most important topics in the field of Math and can be used to solve a variety of problems. Many times one might have come across problems that relate to the prime factorization of a number, to the divisors of a number, to the multiples of a number and so on.

Euler's Totient function is a function that is related to getting the number of numbers that are coprime to a certain number $X$ that are less than or equal to it. In short , for a certain number $X$ we need to find the count of all numbers $Y$ where $gcd(X, Y) = 1$ and $1 \leq Y \leq X$.

A naive method to do so would be to **Brute-Force** the answer by checking the gcd of $X$ and every number less than or equal to $X$ and then incrementing the count whenever a $GCD$ of $1$ is obtained. However, this can be done in a much faster way using Euler's Totient Function.

According to Euler's product formula, the value of the Totient function is below the product over all prime factors of a number. This formula simply states that the value of the Totient function is the product after multiplying the number $N$ by the product of $(1 - (1/p))$ for each prime factor of $N$.

So,

$$\phi(n) = n \prod_{p \text{ prime } p|n} \left(1 - \frac{1}{p}\right)$$

**Algorithm steps:**

- Generate a list of primes.
- While dealing with a certain $N$, check and store all the primes that perfectly divide $N$.
- Now, it is just needed to use these primes and the above formula to get the result.

### Implementation:

```
set<> primes;
static void mark(int num,int max,int[] arr)
{
    int i=2,elem;
    while((elem=(num*i))<=max)
    {
        arr[elem-1]=1;
        i++;
    }
}
GeneratePrimes()
{
    int arr[max_prime];
    for(int i=1;i<arr.length;i++)
    {
        if(arr[i]==0)
        {
            list.add(i+1);
            mark(i+1,arr.length-1,arr);
        }
    }
}
main()
{
    GeneratePrimes();
    int N=nextInt();
    int ans=N;
    for(int k:set)
    {
        if(N%k==0)
        {
            ans*=(1-1/k);
        }
    }
    print(ans);
}
```

There are a few subtle observations that one can make about Euler's Totient Function.

- The sum of all values of Totient Function of all divisors of $N$ is equal to $N$.
- The value of Totient function for a certain prime $P$ will always be $P-1$ as the number $P$ will always have a $GCD$ of $1$ with all numbers less than or equal to it except itself.

- For 2 number A and B, if $GCD(A, B) == 1$ then $Totient(A) \times Totient(B)$ = $Totient(A \cdot B)$.

*Contributed by: Anand Jaisingh*

7
LIVE EVENTS

Did you find this tutorial helpful? YES NO

## TEST YOUR UNDERSTANDING

## Euler Totient Function

In number theory, the totient $\Phi$ of a positive integer $N$ is defined as the number of positive integers less than or equal to $N$ that are co-prime to $N$.

Given an integer $N$. Compute the value of the totient $\Phi$.

**Input:**
First and the only line of input contains single integer $N$.

**Output:**
Print the $\Phi(N)$ in a single line.

**Constraints:**
$1 \leq N \leq 1000000$

SAMPLE INPUT

```
5
```

SAMPLE OUTPUT

```
4
```

Enter your code or Upload your code as file. Save C (gcc 4.8.2) ▼

```c
#include <stdio.h>

int main()
{
    printf("Hello World!\n");
    return 0;
}

```

7 LIVE EVENTS

1:1

Provide custom input

COMPILE & TEST SUBMIT

*Press Ctrl-space for autocomplete suggestions.*

POWERED BY code table

COMMENTS (4)

SORT BY: Relevance▾

Join Discussion...

Cancel Post

**abhishek vanjani** 5 months ago

if k==int 1/k==0 ans=N always,
can someone plz reply

0 votes • Reply • Message • Permalink

**Mouad NACIRI** Edited 5 months ago

Yes you are right, the example code is incorrect :)
Use float
or
Instead of calculating "1 - 1/k" calculate "(k-1)/k" since 1 - 1/k == (k-1)/k
so the formula phi(n) = n * (Product of (k-1)/k) where k is prime
and since all k are divisors of n, u can divide ur n by k and then multiply it by (k-1)
This is an example: http://pastebin.com/KACa2Pz9 (AC in the example problem)

1 vote • Reply • Message • Permalink

**Sushant Gupta** 5 months ago

Euler's Totient function seems to be way slower than using normal method if there is only a single n

▲ 0 votes ● Reply ● Message ● Permalink

**EMILIO WUERGES** 16 days ago

No, because when you only need to generate primes up to sqrt(N).

▲ 0 votes ● Reply ● Message ● Permalink

7

LIVE EVENTS

About Us

Hackathons

Talent Solutions

University Program

Developers Wiki

Blog

Press

Careers

Reach Us

Terms and Conditions | Privacy