

মৌলিক সংখ্যা (**Prime number**) এর অ্যালগরিদম গুলো

Prime number algorithm Bangla tutorial/ Sieve of Eratosthenes



Sharif Hasan

• November 5, 2019

সর্বশেষ আপডেট September 13, 2021

💬 8

🔥 8,755

📖 পড়তে 4 মিনিট লাগতে পারে

অনেকদিন লেখালেখি থেকে দূরে ছিলাম। আজকে লিখতে বসলাম কিন্তু টপিক পাচ্ছিলাম না। ভেবে চিন্তে মনে হলো মৌলিক সংখ্যা বা প্রাইম নাম্বার (**Prime Number**) নিয়ে লিখবো। মৌলিক সংখ্যা হলো বিভিন্ন অ্যালগরিদম এর প্রাণ ভোমরার মতো।

এই লিখারতে কিছু ভুল থাকায় ক্ষমা প্রার্থী। [নতুন লিখায়](#) আমি সব ঠিক করার চেষ্টা করেছি।

প্রাইম ফ্যাক্টরাইজেশন নিয়ে জানতে এটা দেখুনঃ [সংখ্যাতত্ত্ব: মৌলিক সংখ্যা \(prime number\) ও তার অ্যালগরিদম](#)

	2	3	4	5	6	7	8	9	10	Prime numbers
11	12	13	14	15	16	17	18	19	20	
21	22	23	24	25	26	27	28	29	30	
31	32	33	34	35	36	37	38	39	40	
41	42	43	44	45	46	47	48	49	50	
51	52	53	54	55	56	57	58	59	60	
61	62	63	64	65	66	67	68	69	70	
71	72	73	74	75	76	77	78	79	80	
81	82	83	84	85	86	87	88	89	90	
91	92	93	94	95	96	97	98	99	100	
101	102	103	104	105	106	107	108	109	110	
111	112	113	114	115	116	117	118	119	120	

তাহলে, প্রাইম নাম্বার আসলে কি? সহজ হিসাব, ১ এ থেকে বড় যে সংখ্যা কে ১ আর ঐ সংখ্যা ছাড়া অন্য কোন সংখ্যা ছাড়া ভাগ যায় না তাই প্রাইম নাম্বার। এখন যদি বলা

হয় N একটি নাম্বার। আপনাকে বলা হলো এটা প্রাইম কিনা তার একটি প্রোগ্রাম লিখতে।

এর সমাধানটা তো মুটামুটি সংজ্ঞা থেকেই করা যা। একটি ৩ থেকে $N-1$ পর্যন্ত একটা লুপ নিয়ে ঐ সংখ্যা কে ভাগ করে যেতে হবে। যদি ভাগ যায় তবে প্রাইম না। নাহলে প্রাইম নাম্বার। খুব সহজ।

```

1.int i;
2.for(i=3;i<=n-1;i++)
3.{
4.    if(N%i==0)
5.    {
6.        printf("NOT PRIME!");
7.        return 0;
8.    }
9.}
10.printf("PRIME!!");

```

হলো। কিন্তু এখানে একটা সমস্যা আছে। ধরেন সংখ্যাটা 10^{10} থেকে বেশি। তখন আপনি কি করবেন। ওপরে যেই প্রোগ্রামটা দেয়া হয়েছে তার টাইম কমপ্লেক্সিটি $O(N)$ । মানে এটা একটা নাম্বার প্রাইম কিনা তা খুজতে সবচেয়ে বাজে কেসে N বার ঘুরবে। যখন আপনি $N = 10^{10}$ দিবেন তখন 10^{10} বার লুপ ঘুরা অসম্ভব কিছু নয়। কিন্তু যখন আপনাকে সমাধানটা করতে একটা নির্দিষ্ট সময় বেধে দেয়া হবে তখন ?? ধরেন আপনাকে সময় দিল ১ সেকেন্ড। কিন্তু এই ক্ষেত্রে ১ সেকেন্ডের বেশি নিবে উপরের প্রোগ্রাম টা। সুতরাং TLE নিশ্চিত।

এখন এই সমস্যার একটা সমাধান করা দরকার যা আরও কম সময় নিবে। তাহলে কি করা যায়?

এখন, আমরা যদি কোন সংখ্যা $N=50$ এর গুনণীয়ক গুলো খেয়াল করি তবে দেখি একটি গুনণীয়ক d হলে ওপর গুনণীয়ক N/d । সুতরাং আমরা যদি N এর গুনণীয়ক গুলোকে ওপরের শর্তানুযায়ী সাজাই তবে, গুনণীয়ক গুলো হয়,

(1,50) (2,25) (5,10) যার সাধারণ রূপ হলো $(d, N/d)$

লক্ষ করলে দেখা যায় প্রত্যেক জোড়ার ছোট নাম্বারটা সবসময়ই \sqrt{N} এর ছোট বা সমান।

কেন তা প্রমাণ করতে প্রথমে আমরা ধরে নেই দুইটাই \sqrt{N} এর চেয়ে বড়।

সুতরাং,

$a \times b = N$ হওয়ার কথা।

যেখানে a, b দুইটা গুনণীয়ক এবং $a, b > \sqrt{N}$ ।

কিন্তু $\sqrt{N} \times \sqrt{N} = (\sqrt{N})^2 = N$

সুতরাং a, b এর দুইটাই \sqrt{N} এর বড় হতে পারে না। সুতরাং জোড়ার একটা গুণনীয়ককে অবশ্যই \sqrt{N} এর সমান বা ছোট হতে হবে।

সুতরাং ওপরের প্রোগ্রাম এর একটি Efficient রূপ হলো ,

```
1. long long i;  
2.  
3. for(i=2; i<=sqrt(N); i++)  
4.  
5. {  
6.  
7.     if(N%i==0)  
8.  
9.     {  
10.  
11.         printf("NOT PRIME!");  
12.  
13.         return 0;  
14.  
15.     }  
16.  
17. }  
18.  
19. printf("PRIME!!");
```

এই প্রোগ্রামটা \sqrt{N} লুপ ঘুরেই প্রাইম নাম্বার বের করতে পারবে।

কিন্তু এখন আরেকটা সমস্যা এসে দাড়ালো।

যদি বলা হয় ১০০০০০ বার তোমার প্রোগ্রামকে প্রশ্ন করা হবে এবং প্রতিবার তোমাকে একটা নাম্বার দেয়া হবে যেটা হবে 10^7 এর সমান বা ছোট। তবে আপনি কি করবেন? প্রতিবার লুপ ঘুরিয়ে বের করবেন নাম্বারটা প্রাইম কিনা? কিন্তু ধরেন আপনাকে এবার সম দিলো ২ সেকেন্ড। তবুও এবার আপনার TLE খেতে হবে। কারন সবচেয়ে বাজে কেসে আপনার $10^5 \times 10^4 = 10^9$ বার লুপ ঘুরবে। সুতরাং কি করা যায় তবে। এই সমস্যাটা সমাধান করতে হবে যে উপায়ে তার নাম [সিভ অব ইরাটোসথেন্স](#)। এখানে যা করা হয় তা হলো, একটা বুলিয়ান অ্যারে নিবো $N+1$ সাইজের। প্রথমেই

অ্যারের ইলিমেন্টকে ১ করে দিবো। একটা লুপ ঘুরবো $i = 2$ থেকে $i = \sqrt{N}$ পর্যন্ত। এর মাঝখানে আরেকটা লুপ নিবো $j = 2*i$ থেকে $j = N$ পর্যন্ত। মাঝখানের লুপকে ইনক্রিমেন্ট করবো $j += i$ এভাবে। মানে প্রতিবারে j হবে i এর একেকটা গুণিতক। আমরা গুণিতক গুলোকে ০ করে দিবো। এইলুপটা ঢুকার আগে যা করতে হবে তা হলো দেখতে হবে নাম্বার টা আগেই ০ হয়ে গেছে নাকি। যদি আগেই শূন্য হয়ে যায় তবে আর লুপ ঘুরানোর মানে নেইঃ

এখন আপনি যা করবেন তা হলো নাম্বার N ইনপুট নিবেন আর অ্যারে N ইনডেক্সে গিয়ে চেক করবেন যে এটা শূন্য না ১। ১ হলে প্রাইম নাম্বার। না হলে নাই।

```
1. // C++ program to print all primes smaller than
   or equal to
2. // n using Sieve of Eratosthenes
3. #include <bits/stdc++.h>
4. using namespace std;
5.
6. void SieveOfEratosthenes(int n)
7. {
8.     // Create a boolean array "prime[0..n]" and
   initialize
9.     // all entries it as true. A value in
   prime[i] will
10.    // finally be false if i is Not a prime,
   else true.
11.    bool prime[n+1];
12.    memset(prime, true, sizeof(prime));
13.
14.    for (int p=2; p*p<=n; p++)
15.    {
16.        // If prime[p] is not changed, then it
   is a prime
17.        if (prime[p] == true)
18.        {
19.            // Update all multiples of p
   greater than or
20.            // equal to the square of it
21.            // numbers which are multiple of p
   and are
22.            // less than p^2 are already been
   marked.
23.            for (int i=p*p; i<=n; i += p)
24.                prime[i] = false;
25.        }
```

```
26.     }
27.
28.     // Print all prime numbers
29.     for (int p=2; p<=n; p++)
30.         if (prime[p])
31.             cout << p << " ";
32. }
33.
34. // Driver Program to test above function
35. int main()
36. {
37.     int n = 30;
38.     cout << "Following are the prime numbers
        smaller "
39.         << " than or equal to " << n << endl;
40.     SieveOfEratosthenes(n);
41.     return 0;
42. }
```

কোড ক্রেডিট : গিকস ফর গিকস

অতপর অ্যালগরিদম নিয় আরো কিছু লেখা লিখবো । সুতরাং সাথেই থকুন । কোন সমস্যা হলে কमेंট বক্সে কमेंট করুন

আরও পড়ুনঃ [সংখ্যাতত্ত্ব: মডুলার অ্যারিথমেটিক \(Modular arithmetic\); Big mod](#)

লেখাটি কেমন লেগেছে আপনার?

রেটিং দিতে হার্টের উপর ক্লিক করুন।



গড় রেটিং 4 / 5. মোট ভোট: 18

#Number theory

#Prime number

8 টি মন্তব্য

**Unknown**

November 5, 2019 at 9:55 AM

আবার ও শিখলাম, ধন্যবাদ

Reply

**Unknown**

November 5, 2019 at 10:39 AM

Bit field দিয়ে কিভাবে Eratosthenes Sieve C language এ implement করা যায় তার ওপর একটি পোস্ট থাকলে ভালো লাগতো

Reply

**Shanto**

November 5, 2019 at 10:43 AM

নেক্সট আর্টিকলে এটা নিয়ে লিখবো । (:P) (k)

Reply

Pingback: প্রোগ্রামিংঃ টেইল কল রিকার্ন অপটিমাইজেশন টেকনিক । - Sharif Hasan (iishanto)

**Bakibilla baki**

May 27, 2020 at 8:29 PM

শেষের কোডের prime array main() function থেকে ব্যবহার করা যাবে কি??

memset এর complexity কত??

vector prime(n+1,1);

এইটা আর memset এর মধ্যে কোনটা ভালো???

Reply

**io.shanto**

May 28, 2020 at 1:00 AM

হুম, প্রাইম এর মেইন এ ডিক্লেয়ার করা যাবে। সেক্ষেত্রে এরের রেফারেন্স পয়েন্টার আকারে পাঠানো লাগবে।

memset এর কমপ্লেক্সিটি $O(N)$ থেকে বেশি হওয়ার কথা না। vector fill করার কমপ্লেক্সিটিও সমান।

Reply

**Bakibilla baki**

May 28, 2020 at 9:25 AM

ধন্যবাদ

Reply

Pingback: সংখ্যা তত্ত্ব: মৌলিক সংখ্যা ও তার এলগরিদম - শরিফ হাসান