

শাফায়েতের ব্লগ

প্রোগ্রামিং ও অ্যালগোরিদম টিউটোরিয়াল

প্রথম পাতা কনটেন্ট/অ্যালগোরিদম গ্রাফ থিওরি ডাইনামিক প্রোগ্রামিং ডাটা স্ট্রাকচার

প্রাইম জেনারেটর (Sieve of Eratosthenes)

লেখক: শাফায়েত তারিখ: জুলাই ১৭, ২০১১ বিভাগ: [কনটেন্ট প্রোগ্রামিং/প্রবলেম সলভিং](#), [প্রোগ্রামিং](#)

ট্যাগ: [গণিত](#) • [নাম্বার থিওরি](#) • [প্রাইম](#) • [সিভ](#)

প্রাচীনকাল থেকেই গণিতবিদরা মাথা ঘামাচ্ছেন প্রাইম নাম্বার বা মৌলিক সংখ্যা নিয়ে। প্রাইম নাম্বারগুলো মধ্যে লুকিয়ে আছে বিস্ময়কর কিছু সৌন্দর্য। যেকোনো কম্পোজিট বা যৌগিক সংখ্যাকে একাধিক প্রাইমের গুণফল হিসাবে মাত্র একভাবে লেখা যায়, ঠিক যেমন সব যৌগিক পদার্থ একাধিক মৌলিক পদার্থের সমন্বয়ে তৈরি। প্রাচীনকাল থেকেই মানুষ প্রাইম নিয়ে গবেষণা করছে, চলছে এখনো। গাউস, ফার্মা, ইউলারের মত কিংবদন্তি গণিতবিদরা কাজ করেছেন প্রাইম নিয়ে।

1 2 3 4 5 6 7 8 9 10
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

দ্রুত গতিতে প্রাইম সংখ্যা বের করার একটি পদ্ধতি আবিষ্কার করেন Eratosthenes, ২০০ খ্রিস্টপূর্বের একজন গ্রীক গণিতবিদ, বিজ্ঞানি ও কবি। ২২০০ বছরেরও পুরানো সেই পদ্ধতি ব্যবহার করে আমরা আধুনিক কম্পিউটারে প্রাইম জেনারেট করি, খুব কম সময়ে বের করা যায় ১০কোটির নিচে সব প্রাইম সংখ্যা। এই অ্যালগোরিদমটি sieve of Eratosthenes নামে পরিচিত, প্রোগ্রামিং এর জগতে সুন্দরতম অ্যালগোরিদমগুলোর মধ্যে এটি একটি।

sieve এর শাব্দিক অর্থ হলো ছাকনি যা অপ্রয়োজনীয় অংশ ছেটে ফেলে (A sieve, or sifter, separates wanted elements from unwanted material using a woven screen such as a mesh or net)। Eratosthenes এর ছাকনি যৌগিক সংখ্যাগুলোকে ছেটে ফেলে দেয়।

Sift the Twos and sift the Threes,
The Sieve of Eratosthenes.
When the multiples sublime,
The numbers that remain are Prime
(Traditional, collected from wikipedia)

আমরা জানি প্রাইম সংখ্যা হলো সেসব সংখ্যা যাদের ১ এবং সেই সংখ্যাটি ব্যতিত কোনো সংখ্যা দিয়ে ভাগ করা যায়না, যেমন ২, ৩, ৫, ৭, ২৯ ইত্যাদি। অন্যভাবে বলা যায় **সেসব সংখ্যাই প্রাইম যাদেরকে সংখ্যাটির বর্গমূলের সমান বা ছোটো কোনো প্রাইম দিয়ে ভাগ করা যায় না।** এই সংজ্ঞাটাই অ্যালগোরিদমের মূল অংশ, তাই আগে আমরা এটা বুঝতে চেষ্টা করব। ফর্মালভাবে প্রমাণ না করে ব্যপারটি বুঝানোর চেষ্টা করি। যেকোনো সংখ্যাকে আমরা কয়েকটি প্রাইমের গুণফল হিসাবে লিখতে পারি যাদের প্রাইম ফ্যাক্টর বলা হয়:

$$n = p_1 * p_2 * p_3 * \dots * p_k$$

n যদি নিজেই প্রাইম হয় তাহলে $n = p_1 (=n)$ । অন্যথায় অবশ্যই একাধিক প্রাইম ফ্যাক্টর থাকতে হবে। এবার চিন্তা করো কোনো সংখ্যা c কে দুটি সংখ্যার গুণফল $c = a * b$ হিসাবে লিখলে a আর b এর একটি অবশ্যই সংখ্যাটির বর্গমূলের থেকে ছোট, অন্যটি বড়। a, b দুটো সংখ্যাই c এর বর্গমূলের থেকে বড় হলে গুণফল c থেকে বড় হতো (ঠিক যেমন $c = a + b$ হলে a বা b এর একটি c এর অর্ধেকের থেকে ছোট অন্যটি বড়)।

এবার $n = p_1 * p_2 * p_3 * \dots * p_k$ তে ফিরে আসি। p_1, p_2, p_3 ইত্যাদির মধ্যে যে কোনো ২টি যদি n এর বর্গমূল থেকে বড় হয় তাহলে তাদের গুণফল n কে ছাড়িয়ে যাবে, তাই নয় কি? সর্বোচ্চ একটি প্রাইম ফ্যাক্টর বর্গমূলের বাইরে যেতে পারে, বাকিগুলো কে অবশ্যই ভিতরে থাকতে হবে।

তাহলে আমরা নিশ্চিত যে যৌগিক সংখ্যা কে তার বর্গমূলের থেকে ছোট কোনো প্রাইম দিয়ে ভাগ করা যাবে। ২য় সংজ্ঞাটি এখন আমাদের কাছে পরিষ্কার: "সেসব সংখ্যাই প্রাইম যাদেরকে সংখ্যাটির বর্গমূলের সমান বা ছোটো কোনো প্রাইম দিয়ে ভাগ করা যায় না"। বুঝতে না পারলে আরেকবার ভালো করে চিন্তা করে নিচের অংশ পড়ো।

এবার আমরা আমাদের ছাকনি চালু করি এবং প্রাইম বের করি। ২৫ এর নিচের সব প্রাইম আমরা বের করব। ২৫ এর বর্গমূল ৫, তাই ২৫ বা তার থেকে ছোট কোন সংখ্যাকে অবশ্যই ৫ বা তার থেকে ছোট কোনো প্রাইম দিয়ে ভাগ করা যাবে।

২ একটি প্রাইম কারণ ২কে তার বর্গমূলের নিচে কোনো সংখ্যা দিয়ে ভাগ করা যায়না। তাহলে ২ এর মাল্টিপলগুলো কেও প্রাইম নয় কারণ তাদের ২ দিয়ে ভাগ করা যায়, সেগুলোকে আমরা কেটে দেই:

২, ৪, ৬, ৮, ১০, ১২, ১৪, ১৬, ১৮, ২০, ২২, ২৪

২ এর পরের সংখ্যা ৩। ৩ যদি প্রাইম না হতো তাহলে ৩ এর বর্গমূলের নিচের কোনো প্রাইম ৩ কে বাদ দিয়ে দিত, যেহেতু ৩ বাদ পড়েনি তাই সংজ্ঞামতে ৩ প্রাইম। ৩ এর মাল্টিপল গুলো কে বাদ দেই:

৩, ৬, ৯, ১২, ১৫, ১৮, ২১, ২৪

পরের সংখ্যা ৪। ৪ বাদ পড়ে গিয়েছে আগেই। তারপর আছে ৫। ৫ যদি প্রাইম না হতো তাহলে আগেই ছাকনিতে কাটা পড়ত, ৫এর মাল্টিপল গুলোকে কেটে দেই:

৫, ১০, ১৫, ২০, ২৫

আমাদের আর কাটাকাটি প্রয়োজন নেই। ২৫ এর বর্গমূল ৫, তাই ২৫এর নিচের সব সংখ্যার বর্গমূল ৫ থেকে ছোট। সুতরাং ২৫ এর নিচের সকল যৌগিক সংখ্যা ৫ বা তার নিচের কোনো প্রাইম দিয়ে বিভাজ্য। যেহেতু আমরা ২, ৩, ৫ এর সব মাল্টিপল কেটে দিয়েছি, বাকি সংখ্যাগুলো অবশ্যই প্রাইম। ছাকনির উপর থেকে সেগুলো সংগ্রহ করে নেই:

২, ৩, ৫, ৭, ১১, ১৩, ১৭, ১৯, ২৩

আমরা সিভের একটা কোড দেখি:

```
1 bool status[1100002];
2 void siv()
3 {
4     int N=1000000;
5     int sq=sqrt(N);
6     for(int i=4;i<=N;i+=2) status[i]=1;
7     for(int i=3;i<=sq;i+=2){
8         if(status[i]==0)
9             {
10                for(int j=i*i;j<=N;j+=i) status[j]=1;
11            }
12     }
13     status[1]=1;
14
15 }
```

status অ্যারেটা দিয়ে নির্দেশ করে একটি সংখ্যা প্রাইম নাকি কম্পোজিট। status[i]=0 হলে i একটি প্রাইম। শুরুতে সব ইনডেক্সে ০ আছে, আমরা উপরের অ্যালগোরিদম অনুযায়ী নন-প্রাইম সংখ্যা গুলোকে কেটে দিবো, অর্থাৎ j যদি নন-প্রাইম হয় status[j]=1 করে দিবো। ৮ নম্বর লাইনে শুরুতেই ২ এর সব মাল্টিপল কেটে দিলাম। এরপরের পরের লুপটা ৩ থেকে শুরু করে ২ করে বাড়ানোর কারণে জোড় সংখ্যা নিয়ে আর চিন্তা করা দরকার নেই। ১০ নম্বর লাইনে এসে যদি status[i]=0 পাই তাহলে অ্যালগোরিদম অনুযায়ী i অবশ্যই প্রাইম কারণ i এখনও কাটা পড়েনি, এবার i এর সবগুলো মাল্টিপল কেটে দিবো, এজন্য j এর লুপ শুরু করবো 2*i থেকে এবং বাড়ানো i পরিমাণ। আমাদের কাজ শেষ, নন-প্রাইম সংখ্যাগুলো সব কেটে দিবে ভিতরের লুপটি, এখন status[i] এর মান দেখে আমরা i প্রাইম কিনা বের করতে পারবো।


সিভ দিয়ে প্রাইম জেনারেট করে খুব সহজে কোন সংখ্যার প্রাইম ফ্যাক্টর বা উৎপাদকে বিশ্লেষণ করা যায়। এই কাজটা তোমার হাতেই থাকলো :)।

সিভে প্রতিটি সংখ্যা প্রাইম নাকি নন-প্রাইম সেটা আমরা একটি বুলিয়ান অ্যারে দিয়ে চেক করি। যত পর্যন্ত প্রাইম জেনারেট করব তত সাইজের

অ্যারে লাগবে। ১০^৮ আকারের অ্যারে অনেক মেমোরি দখল করে। মেমোরি অপটিমাইজ করার জন্য অসাধারণ একটি পদ্ধতি হলো বিট ব্যবহার করা, একে bitwise সিভ বলা হয়। একটি ইন্টিজারে ৩২টি বিট থাকে যার প্রতিটিকে আমরা ফ্ল্যাগ হিসাবে ব্যবহার করতে পারি, সেটা নিয়ে বিস্তারিত আলোচনা পাবে [এখানে](#)।

ফেসবুকে মন্তব্য

0 comments



Add a comment...

☐ Also post on Facebook

Posting as Duronto Habib
([Change](#))

Comment

Facebook social plugin

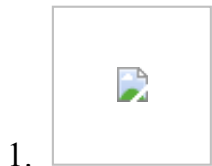
Powered by [Facebook Comments](#)

ট্যাগ: [গণিত](#) • [নাম্বার থিওরী](#) • [প্রাইম](#) • [সিভ](#)

Like A Asif Khan Chowdhury and 33 others like this.

5,851 বার পড়া হয়েছে

১টি মন্তব্য



late_riser

২৩ অক্টোবর ২০১৩

Nice writing.

But, I have a confusion.

In the line 12 (for(int j=i*i;j<=N;j+=i) status[j]=1;) instead of "i*i", won't it be "2*i" ?

Thanks 😊

মন্তব্য করুন(বাংলা ভাষায় বাংলা ফন্টে বা ইংরেজি ভাষায় ইংরেজি ফন্টে)

 নাম(আবশ্যক) ইমেইল(আবশ্যক) ওয়েবসাইট

b *i* ~~del~~ [link](#) b-quote code Close Tags cpp java php

ফোনেটিক বাংলা প্রভাত English

মন্তব্য করুন

(ইংরেজী ফন্টে বাংলা মন্তব্য মুছে ফেলা হতে পারে)

« [ইউভিএ ১০৭০২\(দৌভেলিং সেলসম্যান\)](#)

[জার\(JAR\) ফাইলে ইমেজ/আইকন সমস্যা](#) »

সাবস্কাইব + কানেক্ট

Powered by [OneAll Social Login](#)

ইমেইলে নতুন লেখার আপডেট পেতে উপরের কোনোটির সাথে কানেক্ট করুন অথবা সরাসরি [রেজিস্টার](#) করুন।

আমার সম্পর্কে

শাফায়েত, ঢাকা বিশ্ববিদ্যালয়ের কম্পিউটার বিজ্ঞান বিভাগের ছাত্র। [\(বিস্তারিত...\)](#)



টিউটোরিয়ালের জন্য অনুরোধ!



Follow me on Github



Follow @Shafaet_csedu

113 followers

Goodreads Book Self:



অনুপ্রেরণা:

[কেন আমি প্রোগ্রামিং শিখবো?](#)

কম্পিউটার বিজ্ঞান কেন পড়বো?
প্রোগ্রামিং কনটেন্ট এবং অনলাইন জাজে হাতেখড়ি

অ্যালগোরিদম বেসিক:

বিগ "O" নোটেশন

P-NP,NP-complete,NP-hard ইত্যাদি

গ্রাফ থিওরি:

গ্রাফ থিওরিতে হাতেখড়ি

ভ্যারিয়েবলে গ্রাফ স্টোর করা(ম্যাট্রিক্স ব্যবহার করে)

ভ্যারিয়েবলে গ্রাফ স্টোর করা(লিস্ট ব্যবহার করে)

ব্রুথফোর্স সার্চ (বিএফএস)

মিনিমাম স্প্যানিং ট্রি ১ (প্রিমস অ্যালগো)

মিনিমাম স্প্যানিং ট্রি ২ (ক্রুসকাল অ্যালগো)

টপোলজিকাল সর্ট

ডেপথ ফাস্ট সার্চ এবং আবারো টপোলজিকাল সর্ট

ডায়াক্রিট

স্টেবল ম্যারেজ প্রবলেম

মিনিমাম ভারটেক্স কভার(গ্রাফ+ডিপি)

ট্রি এর ডায়ামিটার নির্ণয়

ডাইনামিক প্রোগ্রামিং:

শুরুর কথা

ডিপি 'স্টেট',NcR, ০-১ ন্যাপস্যাক

কয়েন চেঞ্জ, রক ক্লাইমিং

LIS এবং ডিপি সলিউশন প্রিন্ট করা

বিটমাস্ক ডিপি

মিনিমাম ভারটেক্স কভার(গ্রাফ+ডিপি)

uva 10702 ট্রাভেলিং সেলসম্যান প্রবলেম

ব্যাকট্র্যাকিং:

ব্যাকট্র্যাকিং বেসিক এবং পারমুটেশন জেনারেটর

ডাটা স্ট্রাকচার:

লোয়েস্ট কমন অ্যানসেস্টর (নতুন)

ডিরেঞ্জমেন্ট

মিট ইন দ্যা মিডল (নতুন)


দরকারি সব লিংক এবং ইবুক আর্কাইভ...

ব্লগ আর্কাইভ

ট্যাগ

[LIS MVC nCr](#) [অ্যালগোরিদম](#) [ইউনিয়ন ফাইন্ড](#) [ওয়ার্ডপ্রেস](#) [ওয়েব ডিজাইন](#) [কমপ্লেক্সিটি](#) [কম্প্রেশন](#) [কম্বিনেটোরিক্স](#) [গণিত](#) [গ্রাফ](#) [গ্রাফ থিওরি](#) [জার](#) [ট্রাভেলিং সেলসম্যান](#) [ট্রি](#)
[ডাইনামিক প্রোগ্রামিং](#) [ডাটা স্ট্রাকচার](#) [ডায়াক্রিটা](#) [ডায়ামিটার](#) [ডি-রেঞ্জমেন্ট](#) [ডিএফএস](#) [ডিপি](#) [ডিভাইড](#) [এন্ড কনকোয়ার](#) [ডিরেকশন](#) [অ্যারে](#) [ডিসজয়েন্ট](#)
[সেট](#) [নাম্বার থিওরি](#) [ন্যাপস্যাক](#) [পারমুটেশন](#) [প্রাইম](#) [প্রোগ্রামিং](#) [ফ্রেমওয়ার্ক](#) [বাইপারটাইট](#) [ম্যাচিং](#) [বিএফএস](#) [বিগ O](#) [বিটমাস্ক](#) [ব্র্যাকট্র্যাকিং](#) [ভারটেক্স](#) [কভার](#) [ম্যাচিং](#) [রানটাইম](#) [সি++](#) [সিভ](#)
[সেগমেন্ট ট্রি](#) [স্টেবল ম্যারেজ](#) [স্টেবল সর্ট](#)

କଥୋପକଥନ



[Last Message](#)
1 day, 9 hours ago
4 guests are online.

Info : Please, resolve the addition below before post any new comment...

শাফায়েত : @4315: He don't know me personally :-/, Why don't you send him a mail?

Guest_4315 : Shafaet vai, please request Mr. Felix Halim to make his CP3 book available in Bangladesh. I really need it 😞

শাফায়েত : baccha_coder: গ্রাফ বা ডিপির বেসিক শিখে ফেলো। আমার ব্লগেই পাবে, বা ফেলিক্স হালিমের বই বা অন্য রিসোর্স থেকে শিখতে পারো। আমি সময় করে লেখার চেষ্টা করবো।

শাফায়েত : @EEE2CSE: ল্যাংগুয়েজ শেখা হলে অ্যালগোরিদম শিখতে থাকো আর প্রবলেম সলভ করো, এটা ছাড়া পথ নাই :)।

baccha_coder : Vaiya , I want to be a good programmer. But I don't know where to start . I am practicing Ad-hoc problem. but now I want to solve more. What shall I do? If you write something relating this , It will be better to make a way from where to start practicing and where I can learn a lot about programming.. & always thank you for taking this type of very helpful steps to know about programming very well.

+
-

Captcha:

0+9=

▼

Name:

Guest_1452

Message:

Send

নতুন পোস্ট

- [ডাইনামিক প্রোগ্রামিং: লংগেস্ট কমন সাবসিকোয়েন্স](#)
- [লোয়েস্ট কমন অ্যানসেস্টর](#)
- [গ্রাফ থিওরিতে হাতেখড়ি-৪\(ব্রেথড ফার্স্ট সার্চ\)](#)
- [মিট ইন দ্যা মিডল টেকনিক](#)
- [ট্রি ডায়ামিটার \(অ্যালগোরিদমের প্রমাণ সহ\)](#)

নতুন মন্তব্য

- Mojam Dewan: [প্রোগ্রামিং কনটেন্ট এবং অনলাইন জাজে হাতেখড়ি](#)
- SHAFI-UD-DOULA SOUROV: [দরকারি সব লিংক এবং ইবুক....](#)
- Johne948: [দরকারি সব লিংক এবং ইবুক....](#)
- rason: [গ্রাফ থিওরিতে হাতেখড়ি-৪\(ব্রেথড ফাস্ট সার্চ\)](#)

আদমশুমারি(ডিসেম্বর ১৭,২০১০ হতে)

119518

প্রোগ্রামিং কনটেন্ট ব্লগ

- [acm solver\(আহমেদ শামসুল আরেফিন\)](#)
- [I me and myself\(যোবায়ের হাসান\)](#)
- [I solved a problem\(আরিফুজ্জামান আরিফ\)](#)
- [One problem a day\(সাব্বির ইউসুফ সানি\)](#)
- [Smilitude\(ইকরাম মাহমুদ ফাহিম\)](#)
- [Vexorian](#)
- [প্রোগ্রামিং-বাংলা ইনফরমেটিক্স ব্লগ](#)

প্রিয় কিছু সাইট

- [Notes from SAOS\(সেইন্ট আতিক\)](#)
- [খান একাডেমি](#)
- [ফাহিম ভাইয়ের পাইথন টিউটোরিয়াল](#)
- [শিক্ষক.কম](#)
- [সুবিন ভাইয়ের প্রোগ্রামিং বই](#)

প্রবলেম

[To infinity and Beyond!](#)

[SHAHBG!](#)

টুকিটাকি

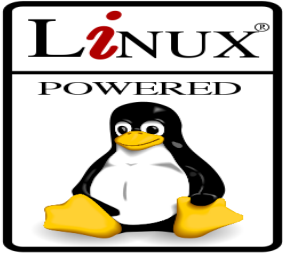
[চিন্তা করার আনন্দ](#)

[বাংলা উইকিপিডিয়া খুঁজছে-আপনাকেই!\(রাগিব হাসান\)](#)

Random Quotes: There is only one success--to be able to spend your life in your own way.

Time Elapsed Since you've loaded this page:

0 Hour 0 Minute 15 Second



Publishing any content of this blog without [author's](#) permission is strictly prohibited.

Powered By [Wordpress](#) Designed By [Ridgey](#) Tweaked and translated in bangla by [Shafaet](#)

Hosting provider: [Mellowhost](#)

Logos and Buttons design: [cooltext.com](#)