

শাফায়েতের ব্লগ

প্রোগ্রামিং ও অ্যালগোরিদম টিউটোরিয়াল

প্রথম পাতা কনটেন্ট/অ্যালগোরিদম গ্রাফ থিওরি ডাইনামিক প্রোগ্রামিং ডাটা স্ট্রাকচার

বিটওয়াইজ সিভ (Bitwise sieve)

লেখক: শাফায়েত তারিখ: নভেম্বর ৫, ২০১১ বিভাগ: [কনটেন্ট প্রোগ্রামিং](#) [প্রবলেম সলভিং](#), [প্রোগ্রামিং](#)

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

বিটওয়াইজ সিভ প্রাইম সংখ্যা বের করার জন্য প্রচলিত অ্যালগোরিদম [Sieve of Eratosthene](#) এ মেমরির ব্যবহার অনেক কমিয়ে আনা যায়! সাধারণ সিভে N পর্যন্ত প্রাইম জেনারেট করলে N সাইজের একটি অ্যারে ডিক্লেয়ার করতে হয়। অ্যারের প্রতিটি এলিমেন্ট একটি করে ফ্ল্যাগ হিসাবে কাজ করে যেটা দেখে আমরা বুঝি একটি সংখ্যা প্রাইম নাকি কম্পোজিট। বিটওয়াইজ সিভে আমরা ফ্ল্যাগ হিসাবে ইন্টিজার বা বুলিয়ান এর বদলে সরাসরি বিট ব্যবহার করবো।

এ টিউটোরিয়াল পড়ার আগে দুটি বিষয় তোমাকে জেনে আসতে হবে

১. Sieve of Eratosthene এর সাধারণ ভার্শন,তুমি এটা আমার [এই পোস্টটি](#) পড়ে শিখতে পারবে সহজেই।

২. সি/সি++ এ বিটওয়াইজ অপারেটরের ব্যবহার। এটাও খুব সহজে শিখতে পারবে [এখান থেকে](#)। টিউটোরিয়ালটির ১ম ২টি অংশ খুব ভালো করে পড়ে ফেলো,বিটমাস্ক ডিপি,বিএফএস যখন শিখবে তখন অনেক কাজে লাগবে।

আশা করি এখন তুমি বিটওয়াইজ অপারেটর সম্পর্কে অনেক কিছু জানো,সাধারণ সিভ লিখতে কোনো সমস্যা হয়না তোমার। এবার আমরা শিখবো বিটওয়াইজ সিভ।

সাধারণ সিভে status বা flag অ্যারেটার কাজ কি? এই অ্যারের ইনডেক্সের মান দেখে আমরা বলতে পারি একটি সংখ্যা প্রাইম কিনা। ধরলাম তোমার status অ্যারেটা ইন্টিজার টাইপের। প্রতিটি ইন্টিজারের মধ্য আছে ৩২টি বিট। আমরা কেনো এতগুলো বিট ব্যবহার করবো খালি ০ বা ১ নির্দেশ করতে? আমরা অ্যারের যেকোনো ইনডেক্সের প্রতিটি বিট দিয়ে একটি সংখ্যা নির্দেশ করতে পারি।

তুমি যখন ইন্টিজার অ্যারেতে ১-৭ পর্যন্ত প্রাইম জেনারেট কর, তোমার অ্যারের অবস্থা বাইনারিতে থাকে এরকম:

```
1 status[১]=০০০.....০০(৩২টি শূন্য)
2 status[২]=০০০.....০১(৩১টি শূন্য,১টি ১)
```

```

3 status[0]=০০০.....০১(৩১টি শূন্য, ১টি ১)
4 status[8]=০০০.....০০(৩২টি শূন্য)
5 status[৫]=০০০.....০১(৩১টি শূন্য, ১টি ১)
6 status[৬]=০০০.....০০(৩২টি শূন্য)
7 status[৭]=০০০.....০১(৩১টি শূন্য, ১টি ১)

```

প্রতিটি ইনডেক্সে ৩১টি বিট কোনো কাজে লাগছেনো অথচ এই বিশাল সংখ্যক অব্যবহৃত বিট আমরা সহজেই কাজে লাগাতে পারি। আমরা ধরে নিবো:

```

1 status[০] এর
2     >>> শূন্যতম বিট ০ এর প্রাইমালিটি নির্দেশ করে (সবথেকে ডানের বিট==০ তম বিট)
3     >>> ১ম বিট ১ এর প্রাইমালিটি নির্দেশ করে (সবথেকে ডানের বিট==০ তম বিট)
4     >>> ২য় বিট ২ এর প্রাইমালিটি নির্দেশ করে
5     >>> ৩য় বিট ৩ এর প্রাইমালিটি নির্দেশ করে
6     .....
7     >>> ৩১তম বিট ৩১ এর প্রাইমালিটি নির্দেশ করে
8
9 status[১] এর
10    >>> শূন্যতম বিট ৩২ এর প্রাইমালিটি নির্দেশ করে
11    >>> ১ম বিট ৩৩ এর প্রাইমালিটি নির্দেশ করে
12    .....
13
14 status[২] এর
15    >>> শূন্যতম বিট ৬৪ এর প্রাইমালিটি নির্দেশ করে
16    >>> ১ম বিট ৬৫ এর প্রাইমালিটি নির্দেশ করে

```

তাহলে ১-৭ পর্যন্ত প্রাইম জেনারেট করলে তোমার অ্যারের অবস্থা দাড়াবে:

status[১]= ০০০০.....১০১০১১০০(মোট ৩২টি বিট) (সবথেকে ডানের বিট==০ তম বিট)

৭টি সংখ্যার কাজ একটি ইনডেক্সেই শেষ!!। শুধু ৭টি নয়, আসলে ৩১টি সংখ্যার কাজ শেষ হবে ১টি ইনডেক্স(কারণ প্রতি ইনডেক্সের ৩১টি বিট ব্যবহার করছি আমরা)। এখন প্রশ্ন হলো কোনো সংখ্যার প্রাইমালিটি কত নম্বর ইনডেক্সের কত নম্বর বিট দিয়ে নির্দেশ করা হবে? খুব সহজ, সংখ্যাটি i হলে i/৩২ নম্বর ইনডেক্সের i%৩২ নম্বর বিট আমাদের চেক করতে হবে। তাহলে i=১ হলে চেক করবো ০ নম্বর ইনডেক্সের ১ নম্বর বিট, i=৩৩ হলে চেক করবো ১ নম্বর ইনডেক্সের ১ নম্বর বিট ইত্যাদি। (শূন্য বেসড ইনডেক্সিং)

কোডিং অংশ একদম সহজ। তুমি যেহেতু বিটওয়াজ অপারেটরের ব্যবহার জানো, কোনো সংখ্যার pos তম বিটে ১ বা ০ আছে নাকি সহজেই চেক করতে পারবে। pos তম বিটে নিজের ইচ্ছামত ১ বা ০ বসাতেই পারবে। আমাদের এখানে ০ বসানো দরকার নেই, ১ বসাতে পারলেই চলবে। দুটি ফাংশন লিখে ফেলি:

```

1 bool Check(int N,int pos){return (bool)(N & (1<<pos));}
2 int Set(int N,int pos){    return N=N | (1<<pos);}

```

এবার সিভ লিখে ফেলি:

```
1 int N =100,prime[100];
2 int status[100/32];
3 void sieve()
4 {
5     int i, j, sqrtN;
6     sqrtN = int( sqrt( N ) );
7     for( i = 3; i <= sqrtN; i += 2 )
8     {
9         if( check(status[i/32],i%32)==0)
10        {
11            for( j = i*i; j <= N; j += 2*i )
12            {
13                status[j/32]=Set(status[j/32],j % 32) ;
14            }
15        }
16    }
17    puts("2");
18    for(i=3;i<=N;i+=2)
19        if( check(status[i/32],i%32)==0)
20            printf("%d\n",i);
21
22 }
```

লক্ষ্য করো,আমরা সাধারণ সিভের মত করেই সব লিখেছি,তবে status[i] এর মান চেক করার বদলে status[i/32] এর i%32 নম্বর বিটের মান চেক করেছি।

বিটওয়াইজ সিভ ব্যবহার করে 10^8 পর্যন্ত প্রাইম তুমি জেনারেট করে ফেলতে পারবে। সাধারণ সিভের থেকে সময় + মেমরি কম লাগবে। সাধারণ গুণ,ভাগ অপারেশনের থেকে বিটের অপারেশনগুলো দ্রুত কাজ করে। আমরা আরো কিছু অপটিমাইজেশন করতে পারি। যেমন তুমি উপরে দেয়া [টিউটোরিয়াল](#) পড়ে থাকলে জানো যে কাওকে ২ দিয়ে গুণ করা আর সংখ্যাটির বাইনারিকে ১ ঘর বামে শিফট করা একই কথা। আবার ২ দিয়ে ভাগ করা আর ১ ঘর ডানে শিফট করা একই কথা,৩২ দিয়ে mod করা আর ৩১ দিয়ে AND করা একই কথা। তাহলে আমরা নিচের মত করে কোডটি লিখতে পারি:

```
1 int status[(mx/32)+2];
2 void sieve()
3 {
4     int i, j, sqrtN;
5     sqrtN = int( sqrt( N ) );
6     for( i = 3; i <= sqrtN; i += 2 )
7     {
8         if( Check(status[i>>5],i&31)==0)
9         {
10            for( j = i*i; j <= N; j += (i<<1) )
```

```

11         {
12             status[j>>5]=Set(status[j>>5],j & 31) ;
13         }
14     }
15 }
16
17 puts("2");
18 for(i=3;i<=N;i+=2)
19     if( Check(status[i>>5],i&31)==0)
20         printf("%d\n",i);
21 }

```

ফাংশন ব্যবহার না করে ম্যাক্রো ব্যবহার করলে আরো কম সময় লাগবে। [প্রোগ্রামিং কনটেন্টে](#) খুব কমই বিটওয়াইজ সিভ ব্যবহার করা দরকার হয়,সাধারণ সিভেই কাজ চলে। তারপরেও এটা শিখলে বিটের কনসেপ্ট গুলো কিছুটা পরিষ্কার হবে,অন্য কোনো প্রবলেমে হয়তো মেমরি কমিয়ে ফেলতে পারবে। ডাইনামিক প্রোগ্রামিং এ আমরা প্রায়ই বিটমাস্কের ব্যবহার করি মূলত মেমরি কমানোর জন্য।

তোমার ইম্প্লিমেন্টেশন সঠিক নাকি চেক করতে নিচের সমস্যাটি সমাধান করে ফেলো:

<http://www.spoj.pl/problems/TDPRIMES/>

(নোট: অনেকের ভুল ধারণা আছে যে bool টাইপের ভ্যারিয়েবলের আকার ১বিট। আসলে bool এর আকার ৮বিট বা এক বাইট,char এর সমান। এর কারণ হলো কম্পিউটার ১ বাইটের ছোটো মেমরি সেগমেন্টকে অ্যাড্রেস করতে পারেনা, তাই ভ্যারিয়েবলের ন্যূনতম আকার ১ বাইট)

ফেসবুকে মন্তব্য

6 comments

6 comments

[Add a comment](#)



Imran Chowdhury Emi · [Follow](#) · Dhaka, Bangladesh · 1,136 subscribers

tnx

[Reply](#) · [Like](#) · [Follow Post](#) · 10 December 2013 at 12:56



Saiful Barih Maruf · [Follow](#) · Works at Microsoft Student Partners Bangladesh

ম্যাক্রো এর ব্যবহারটা পারতেছি না ভাইয়া... একটু সাহায্য দরকার.....।.

[Reply](#) · [Like](#) · [1](#) · [Follow Post](#) · 10 June 2013 at 13:16



Rafid Mustafiz · Study at BSC(Engg.)in CSE at Student অনেক উপকার হইল via.tnxs a lot.

[Reply](#) · [Like](#) · [1](#) · [Follow Post](#) · 12 March 2013 at 09:11



Abdulla Al Sun · [Follow](#) · Khulna University of Engineering and Technology

প্রথম জানলাম বিট ওয়াইজ সিভ.

[Reply](#) · [Like](#) · [1](#) · [Follow Post](#) · 26 December 2012 at 02:24



আব্দুল্লাহ আল নোমান · [Follow](#) · Works at Managing director of toto company

ভাল হইছে tnxs vaia...

[Reply](#) · [Like](#) · [3](#) · [Follow Post](#) · 23 May 2012 at 16:35



Nuruzzaman Milon · [Follow](#) · Top commenter · Software Engineer at Brotecs Technologies Limited

চমৎকার।

[Reply](#) · [Like](#) · [2](#) · [Follow Post](#) · 13 April 2012 at 10:58

Facebook social plugin

Powered by [Facebook Comments](#)

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

Like 42 people like this.

4,356 বার পড়া হয়েছে

৯টি মন্তব্য

1.



হাসান

১৫ আগস্ট ২০১২

ভাইয়া, খুবই দারুণ একটা পোস্ট।

UVA 10311 (Goldbach and Euler) এবার করে ফেলা যাবে!

০



শাফায়েত

১৫ আগস্ট ২০১২

ধন্যবাদ। 10311 এ অবশ্য bitwise sieve দরকার হয়নি আমার।

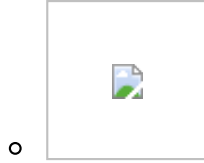
2.



অনিন্দ্য

২৩ অক্টোবর ২০১২

ভাই, ১৫ তম লাইনে $j\%31$ এর পরিবর্তে $j\%32$ হবে না? $j\%32$ এবং $j\&31$ একই হবার কথা। যদি $P=2^x$ হয়, তাহলে $n\%p = n\&(p-1)$ হওয়ার কথা। এটা নিয়ে একটু confusion হচ্ছে। 😞



শাফায়েত

২৩ অক্টোবর ২০১২

হ্যা $j\%32$ হবে, আমি ঠিক করে দিচ্ছি। যদিও $j\%31$ লিখলেও কোডটা কাজ করে এজন্য হয়তো খেয়াল করিনি ঠিকমত তাই কনফিউশন তৈরি হয়েছে।



3.

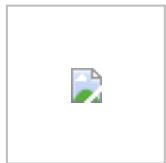
[কাইদুল](#)

২৭ অক্টোবর ২০১২

int(signed integer) এর left Most bit যেহেতু sign bit, তাই আমরা যখন একে left shifting করি তখন left most বিটে 1 পুশ করলে সংখ্যাটা নেগেটিভ হওয়ার সম্ভাবনা থাকে। তাই $i < 1$ ব্যবহার করা কি ঠিক হবে?

আমি ভুল হতে পারি, for($j = i*i$; $j \leq N$; $j += 2*i$) লাইনের মাধ্যমে কি সব মাল্টিপল বের হবে? for($j = 2*i$; $j \leq N$; $j += i$) দিলে হবার কথা!

$i\%31$ গুলোকে $i\%32$ দিয়ে রিপ্লেস করলে ভাল হত। কিছুটা কনফিউজিং লাগছে 😞



4.

শাফায়েত

৩০ অক্টোবর ২০১২

টেকনিক্যাল কিছু সমস্যার জন্য কোড এডিট করতে পারছি না, ঠিক হলে রিপ্লেস করে দিবো।
i*1 দিলেও কাজ হবে, http://www.lightoj.com/article_show.php?article=1001 এই লিংকটা দেখেন।

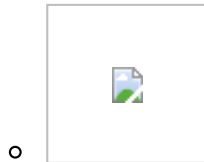


5.

আলিম

১১ এপ্রিল ২০১৩

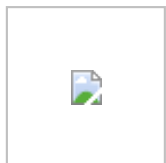
ভাইয়া, এই প্রবলেম এর ক্ষেত্রে কিভাবে প্রাইম Generate করব?
<http://www.spoj.com/problems/KPRIMES2/>



শাফায়েত

১২ এপ্রিল ২০১৩

এটায় মনে হচ্ছে আরো অ্যাডভান্সড কিছু লাগবে, এই থ্রেডটা দেখো : <http://apps.topcoder.com/forums/?module=Thread&threadID=698955&start=0&mc=5#1335192>



6.

Maruf Tuhin

৮ ডিসেম্বর ২০১৩

583 – Prime Factors

For this problem i got 0.912 runtime for normal sieve. <http://ideone.com/CoWXez>

I converted the sieve portion to bitwise sieve, and got 2.485 runtime!!!! <http://ideone.com/HH3LwY>

Why????????????????

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)

« [গ্রাফ থিওরিতে হাতেখড়ি ৭:টপোলজিকাল সর্ট](#)

[সফটওয়্যার আর্কিটেকচার: মডেল-ভিউ-কন্ট্রোলার \(MVC\) »](#)

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)
ব্রেথড ফার্স্ট সার্চ (বিএফএস)
মিনিমাম স্প্যানিং ট্রি ১ (প্রিমস অ্যালগো)
মিনিমাম স্প্যানিং ট্রি ২ (ক্রুসকাল অ্যালগো)
টপোলজিকাল সর্ট
ডেপথ ফার্স্ট সার্চ এবং আবারো টপোলজিকাল সর্ট
ডায়াক্রিটা
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 ক্লাইম্বিং
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:

[সেগমেন্ট ট্রি-১](#)

[সেগমেন্ট ট্রি-২\(লেজি প্রপাগেশন\)](#)

[ডিসজয়েন্ট সেট\(ইউনিয়ন ফাইন্ড\)](#)

[অ্যারে কমপ্ৰেশন/ম্যাপিং](#)

[ট্রাই\(প্রিফিক্স ট্রি/রেডিক্স ট্রি\)](#)

[লোয়েস্ট কমন অ্যানসেস্টর\(নতুন\)](#)

নাম্বার থিওরি/গণিত:

[মডুলার অ্যারিথমেটিক](#)

[প্রাইম জেনারেটর \(Sieve of Eratosthenes\)](#)

[বিটওয়াইজ সিভ](#)

[ডিরেঞ্জমেন্ট](#)

অন্যান্য :

[ডিরেকশন অ্যারে](#)

[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](#) [ডায়ালগ বক্স](#) [ডায়ালগ বক্স](#) [ডি-রেঞ্জমেন্ট](#) [ডিএফএস](#) [ডিপি](#) [ডিভাইড এন্ড কনকোয়ার](#) [ডিরেকশন অ্যারে](#) [ডিসজয়েন্ট সেট](#) [নাম্বার থিওরি](#) [ন্যাপস্যাফ](#) [পারমুটেশন](#) [প্রাইম প্রোগ্রামিং](#) [ফ্রেমওয়ার্ক](#) [বাইপারটাইট ম্যাচিং](#) [বিএফএস](#) [বিগ O](#) [বিটমাস্ক](#) [ব্যাংকট্র্যাফিকিং](#) [ভারটেক্স কভার](#) [ম্যাচিং](#) [রানটাইম সি++](#) [সিভ](#) [সেগমেন্ট ট্রি](#) [স্টেবল ম্যারেজ](#) [স্টেবল সর্ট](#)



কথোপকথন



Last Message
1 day, 9 hours ago
1 guest is 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:

9+8=

Name:

Guest_433

Message:

Send

নতুন পোস্ট

- [ডাইনামিক প্রোগ্রামিং: লংগেস্ট কমন সাবসিকোয়েন্স](#)
- [লোয়েস্ট কমন অ্যানসেস্টর](#)
- [গ্রাফ থিওরিতে হাতেখড়ি-৪\(ব্রেথড ফাস্ট সার্চ\)](#)
- [মিট ইন দ্য 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....](#)
- razon: [গ্রাফ থিওরিতে হাতেখড়ি-৪\(ব্রেথড ফাস্ট সার্চ\)](#)

আদমশুমারি(ডিসেম্বর ১৭,২০১০ হতে)

১১৪৫১৪

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

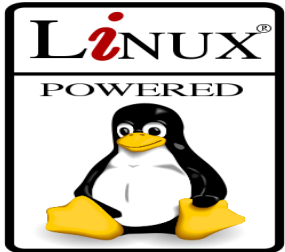
চিন্তা করার আনন্দ

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 17 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](#)