প্রোগ্রামিং - Programming   সংখ্যাতত্ত্ব - Number theory

সংখ্যাতত্ত্ব: ইউক্লিডিয়ান অ্যালগরিদম ও গ.সা.গু

Euclidean algorithm Bangla tutorial/ Euclidean GCD and LCM/ O(log n) এ গ.সা.গু. নির্নয়

Sharif Hasan • August 7, 2021 সর্বশেষ আপডেট August 25, 2021 4 541 পড়তে 2 মিনিট লাগতে পারে

ইউক্লিডিয়ান অ্যালগরিদম হলো গ.সা.গু. বা গরিষ্ঠ সাধারণ গুণনীয়ক বের করার জন্য একটি দ্রুতগতির অ্যালগরিদম। এই অ্যালগরিদমের নামকরণ করা হয় ইউক্লিডের নামে যিনি তার Elements নামক বইয়ে বর্ননা করেছেন,

দুটি সংখ্যার গ.সা.গু. হল সেই বৃহত্তম সংখ্যা যা দুটি সংখ্যাকেই নিঃশেষে ভাগ করে। ইউক্লিডীয় অ্যালগরিদম যে নীতির উপর প্রতিষ্ঠিত তা হলো বৃহত্তর সংখ্যাটি থেকে ক্ষুদ্রতর সংখ্যাটি বিয়োগ করা হলেও তাদের গ.সা.গু. পরিবর্তিত হয় না। যেমন আমরা যদি দুইটি সংখ্যা ২৫০ এবং ১০৫ নেই যাদের গ.সা.গু. ৫ । ২৫০ থেকে ১০৫ বাদ দিলে পাই, ১৪৫। এখানে ১০৫ এবং ১৪৫ এর গসাগু ও ৫ হয়। এখন ১৪৫ থেকে ১০৫ বাদ দিলে পাবো ৪০। ১০৫ এবং ৪০ এর গ.সা.গু. ও ৫। এখন ১০৫ থেকে ৪০ বাদ দিলে পাবো ৬৫, ৬৫ থেকে ৪০ বাদ দিলে পাবো ২৫, ৪০ থেকে ২৫ বাদ দিলে ১৫, ২৫ থেকে ১৫ বাদ দিলে ১০, ১৫ থেকে ১০ বাদ দিলে ৫ এখন ১০ থেকে ৫ বাদ দিলে থাকে ৫। যা আমাদের গ.সা.গু.।

অর্থাৎ যতক্ষণ পর্যন্ত ছোট সংখ্যা আর বড় সংখ্যা সমান না হয় আমরা পর্যায়ক্রমিক বিয়োগ চালাতে থাকবো। অনেকটা নিচের মতো।

```
1  int gcd(int a,int b){
2   while(a!=b){
3   if(a>b){
4   a-=b; //a বড় হলে a-b দিয়ে a কে আপডেট করবো।
5   }else{
6   b-=a; //b বড় হলে b-a দিয়ে b কে আপডেট করবো।
7   }
8   }
9   return a;
10 }
```

এখন উপরের কাজটিতে গসাগু বের করা গেলেও এতে সময় বেশি লাগবে। আমরা ইচ্ছে করলেই ভাগ করার মাধ্যমে এই প্রসেসকে O(log n) সময়ে নামিয়ে আনতে পারি। খেয়াল করি, ১০৫ এবং ২৫০ এর গসাগু বের করার সময় ১০৫ কে পর পর দুইবার ২৫০, ১৪৫ থেকে বিয়োগ দিতে হয়েছিলো। কেমন হয় যদি আমরা ভাগ এর মাধ্যমে করি? ১৪৫ থেকে ১০৫ বাদ দিলে পেয়েছিলাম ৪০। $250\%105 = 40$ সুতরাং আমরা যদি ৪ এবং ৬ নং লাইনে বিয়োগ এর বদল ভাগশেষ (Modulo or %) বের করি

তাহলেই ২ বারের কাজ ১ বারে হয়ে যাচ্ছে।

```
int gcd_iterative(int a,int b){
 while(a!=0&&b!=0){
 if(a>b){
 a%=b;
 }else{
 b%=a;
 }
 }
 return max(a,b);
}
```

এখানে কয়েকধাপের বিয়োগের কাজ একবারেই সম্পন্ন হবে। যার জন্য আমরা O(log n) সময়ে ভাগশেষ বের করতে পারবো। যেহেতু ভাগশেষ বের করতেছি প্রতিবার, তার কোন না কোন সময় a অথবা b এর কোন একটি শুন্য হবে। তখন আমাদের লুপ থামবে এবং a, b এর মধ্যে যা অশূন্য তাকে রিটার্ন করবো।

উপরের কোডে আমরা একটি বিষয় খেয়াল করলাম। যেই সংখ্যাটি দিয়ে ভাগশেষ বের করা হচ্ছে তা সবসময় অন্যটির থেকে ছোট। এই আইডিয়া কাজে লাগিয়ে আমরা রিকার্শনের মাধ্যমে গ.সা.গু. বের করতে পারি।

রিকার্শনের মাধ্যমে গ.সা.গু. নির্ণয়

ধাপ	a	b	$gcd(a,b)$	Update a to b	Update b to $a\%b$
১	250	105	$gcd(250,105)$	$a=105$	$b=250\%105=40$
২	105	40	$gcd(105,40)$	$a=40$	$b=105\%40=25$
৩	40	25	$gcd(40,25)$	$a=25$	$b=40\%25=15$
৪	25	15	$gcd(25,15)$	$a=15$	$b=25\%15=10$
৫	15	10	$gcd(15,10)$	$a=10$	$b=15\%10=5$
৬	10	5	$gcd(10,5)$	$a=5$	$b=10\%5=0$
৭	5	0	যেহেতু $b=0$ তাই বলা যায় এর আগের বার b দ্বারা a কে ভাগ গিয়েছিলো। বর্তমানে a তখনকার b কে		

			জমা রেখেছে। তাই a রিটার্ন করবো।		

উপরের ছকটি খেয়াল করি। আমরা ২৫০ এবং ১০৫ এর গসাগু বের করবো। এখানে $b \leq a$। এখন, উপরের আলোচনা থেকে জানি, $gcd(a, b) = gcd(b, a\%b)$, এখানে $b \geq a\%b$। সুতরাং $gcd(a, b)$ বের করতে হলে আমরা আগে $gcd(b, a\%b)$ বের করবো। এভাবে কোন এক সময় b দ্বারা a কে ভাগ করা যাবে। উপরের ৬ নং ধাপে যা হয়েছে। এসময় a আপডেট হয়ে b হবে এবং $b = 0$ হবে (যেহেতু $a\%b = 0$)। তাই ৭ নং ধাপে $b = 0$ বেস কেসে a কে GCD বা গ.সা.গু. হিসেবে রিটার্ন করেছি।

রিকার্শনের মাধ্যমে গ.সা.গু. নির্ণয়ের **C++** কোড

```
1 int gcd_recursion(int a,int b){
2  if(b==0) return a; // যখন b=0 হবে, তখন a আমদের গ.সা.গু.।
3  return gcd_recursion(b,a%b); //a=b এবং b=a%b করে দিলাম রিকার্সিভ ফাংশন কলের মা
4 }
```

৩ নং লাইনে আমরা উপরে যেভাবে বর্ণনা করেছি সেভাবেই রিকার্সিভ কল করেছি, এর মাধ্যমে প্যারামিটারে $a = b$ এবং $b = a$ পাঠিয়েছি। এভাবে যখনি ২ নং লাইনে b=0 পেয়েছি, a কে গ.সা.গু. হিসেবে রিটার্ন করেছি। কারণ 0 এবং a এর গ.সা.গু. a ই হবে।

এই কোডের টাইম কমপ্লেক্সিটি O(log n) যেখানে $n = max(a, b)$।

গ.সা.গু. থেকে ল.সা.গু. **(LCM)** নির্ণয়

আমরা যদি O(log n) এ দুইটি সংখ্যার গ.সা.গু. বের করতে পারি, তবে খুব সহজেই ল.সা.গু. (Least Common Multiple) বের করা যায়। আমরা জানি,

$gcd(a, b) \times lcm(a, b) = a \times b$, যেখানে a, b হলো প্রদত্ত দুইটি সংখ্যা।
বা, $lcm(a, b) = \dfrac{a \times b}{gcd(a, b)}$

অর্থাৎ আমরা যদি দুইটি সংখ্যার গুণফলকে দুইটি সংখ্যার গ.সা.গু. দিয়ে ভাগ করি তবে আমরা $O(\log n)$ এ ল.সা.গু. পাবো।

আজ এই পর্যন্তই। পরের লিখাটি Extended Euclidean algorithm নিয়ে । পরে দেখার নিমন্ত্রন রইলো। #happy_coding

লেখাটি কেমন লেগেছে আপনার?

রেটিং দিতে হার্টের উপর ক্লিক করুন।

গড় রেটিং 4.3 / 5. মোট ভোট: 25

#সংখ্যাতত্ত্ব

4 টি মন্তব্য

Akib Rahman

August 23, 2021 at 12:19 PM

Nice explanation. Thank you so much

Reply

Sharif Hasan

August 23, 2021 at 12:23 PM

💖

Reply

---

guEsS

September 6, 2021 at 10:58 PM

Thanks for your contribution to create my interest about it.

Reply

Sharif Hasan

September 6, 2021 at 11:03 PM

My pleasure ❤️

Reply