



## বিন্যাস করা যাক (পর্ব: ১)

আজ একটু বিন্যাস নিয়ে কথা বলতে ইচ্ছা করছে। বিন্যাস তথা **permutation** সম্পর্কে তো সবারই ধারণা আছে কম-বেশি। আসলে এটা একটু বড় টপিক, তাই আরও দু-একটা আর্টিকলে ডিটেইলস আলোচনা করবো। আজ একটু বিন্যাস কি এবং কিভাবে কোডিং করা যায় তার বেসিক লেভেলের আইডিয়া দিবো। যারা জানো তাদেরকে দূরে থাকার জন্য হুকুম করতেছি।

তো কাজে ফিরে আসি।

### বিন্যাস কি?

File:Permutations RGB.svg

বিন্যাস হলো খুব সহজ ভাষায় “সাজানো”। উপরের ছবিটা উইকি থেকে নেয়া, দেখতেই পাচ্ছে লাল, সবুজ, নীল তিনটা বলকে কতভাবে সাজাইতে পারি আমরা। এখন প্রশ্ন হলো, সাজানোর সময় আমরা কি করি? ধরো, টেবিলের উপর তুমি বই-খাতা রাখতেছো, তো ফিজিক্স প্রথম



### Archives

- October 2013
- July 2013
- December 2012
- May 2012

### Recent Posts

- বিন্যাস করা যাক (পর্ব: ২)
- বিন্যাস করা যাক (পর্ব: ১)
- Chinese Remainder Theorem
- খাতা-কলমে Extended Euclid Method
- Extended Euclidean Algorithm এবং একটুখানি Modular Multiplicative Inverse

### Recent Comments

- ops on বিন্যাস করা যাক (পর্ব: ১)

পত্রের উপর রাখলে কেমিস্ট্রি দ্বিতীয় পত্র, তার উপর রাখলে আবার ফিজিক্স দ্বিতীয়পত্র। তো জিনিসটা একটু অগোছালো হয়ে গেলো বলে তুমি আবার সেটা ঠিক করে ফিজিক্স প্রথমপত্রের পর ফিজিক্স দ্বিতীয়পত্র রাখলে। তার উপর রাখলে কেমিস্ট্রি প্রথম, তারপর দ্বিতীয়... এভাবে। তো এইভাবে অনেক উপায়েই তুমি ওই বইগুলো সাজাতে পারবে। যতভাবেই সাজাতে পারবে এগুলো সবই বিন্যাস করা। অর্থাৎ বিন্যাস করার জন্য একই জিনিসগুলো বারবার ব্যবহৃত হতে পারে তবে সেটা ভিন্ন অর্ডারে সাজানো থাকতে হবে। মনে করো 'A' এবং 'B' দুইটা বই। এগুলো তুমি কতভাবে সাজাতে পারবে? অবশ্যই দুইটি উপায়েই:

1	AB
2	
3	BA

আর কোনো উপায়ে সাজানো যাওয়ার কথা না। এবার ধরো, 'A', 'B', 'C' তিনটা সাজাতে চাচ্ছে। তাইলে কতভাবে সাজাতে পারবে? দেখি আমরা সেটা:

1	ABC
2	
3	ACB
4	
5	BAC
6	
7	BCA
8	
9	CAB
10	
11	CBA

▪ Muhammad Minhazul Haque on বিন্যাস করা যাক (পর্ব: ২)

▪ Duronto Habib on বিন্যাস করা যাক (পর্ব: ১)

▪ Abu Asif Khan Chowdhury on Chinese

Remainder Theorem

▪ TripleM Zim on Chinese Remainder

Theorem

## Blog Traffic

## Pages

Pages | Hits | Unique

▪ Last 24 hours: 75

▪ Last 7 days: 520

▪ Last 30 days: 835

▪ Online now: 1

## Get Updates

Join 1 other subscriber

Email Address

Subscribe

## Meta

▪ Log in

▪ Entries [RSS](#)

আমার মাথায় আর একটাও আসতেছে না এগুলো ছাড়া। দেখতেই পাচ্ছে তিনটা বই তুমি মোট ছয়টা উপায়ে সাজাতে পারতেছে। আসলে যেকোনো সংখ্যক জিনিসই তুমি সাজাতে পারবে সর্বাধিক তার ফ্যাকটোরিয়াল সংখ্যক উপায়ে। অর্থাৎ  $n$  টা জিনিস তুমি সাজাতে পারবে  $n!$  উপায়ে। বিশ্বাস হচ্ছে না আমার কথা? চলো তাইলে তোমাকে একটা প্রমাণ দি। C++ এর `algorithm` ক্লাসের একটা মেথড আছে `next_permutation()`। এটা তোমাকে সম্ভাব্য সকল বিন্যাস “Lexicographical order” (হুম, আমাকেও কয়েকবার পড়তে হয়েছিলো এটা সঠিকভাবে উচ্চারণের জন্য) `generate` করে দিতে পারবে। কোডটা দিলাম, আপাতত কোডের বিভিন্ন জিনিস চেনজ করে এবং ঘাটাঘাটি করে মজা নিতে থাকো। আর “লেক্সিকোগ্রাফিক্যাল অর্ডার” সম্পর্কে একটু একটু আইডিয়া পাইতে পারবে নিচের কোডটার আউটপুট গবেষণা করলে। তাছাড়াও সামনের পর্বগুলোতে একটু একটু আলোচনা করা হবেই এটা নিয়ে, আমরা প্রোগ্রামে `permutation generate` করার সময় এই অর্ডারটাই ফলো করি।

### STL ব্যবহার করে কোড করা:

```
1 #include <iostream>
2 #include <stdio.h>
3 #include <algorithm>
4
5 using namespace std;
6
7 int main ()
8 {
9     char books[] = {'A','B','C','D','E'};
10    int size = 5;
11    printf("Boigula ei upaygulate sajate parbe:\n");
12    int upay = 1;
13    do{
14        printf("%d: ", upay++);
15        for(int i=0; i<size; i++)
16            printf("%c ", books[i]);
17        printf("\n");
18    }
19    while ( next_permutation(books,books+size) );
20    return 0;
21 }
```

পরের পর্বে কিন্তু কিভাবে `next_permutation()` ফাংশনটা তৈরী করা যায় সেটা শেখাবো। ভয় পেয়ো না, এটা তৈরী করা হালুয়া খাওয়ার মতই সোজা যদি রিকার্সিভ ব্যাকট্রাকিং সম্পর্কে হালকা-পাতলা ধারণা থেকে থাকে। Keep coding... (চলবে....)


■ [Comments RSS](#)

■ [WordPress.org](#)

 98 total views, 1 views today

Share this:



 *Lexicographical order , Permutation , Standard Template Library*

◀ Chinese Remainder Theorem

বিন্যাস করা যাক (পর্ব: ২) ▶

*Duronto Habib*

Osam.....

*ops*

recursive backtracking কি জিনিস

^ TOP

আসিফের হ-য-ব-র-ল

Powered by WordPress 3.7 and Theme Mflat <!--91 queries. 1.884 seconds. --!>