[**অ্যারে কম্প্রেশন**](http://www.shafaetsplanet.com/planetcoding/?p=1388)

লেখক: শাফায়েত

খুবই সহজ কিন্তু দরকারি একটা টেকনিক নিয়ে আলোচনা করবো আজকে। STL এর ম্যাপ ব্যবহার করে আমরা অ্যারে কম্প্রেশন করবো। মনে করো কোনো একটা প্রবলেমে তোমাকে বলা হলো একটি অ্যারেতে ১ লাখটা সংখ্যা দেয়া থাকবে যাদের মান হবে ০ থেকে সর্বোচ্চ ১০০০। এখন যেকোনো আমি একটি সংখ্যা বললে তোমাকে বলতে হবে অ্যারের কোন কোন পজিশনে সংখ্যাটি আছে। যেমন মনে করো ইনপুট অ্যারেটা হলো:

১ ০ ০ ২ ৫ ২ ১ ০ ৪ ৫ ১ ২

খুবই সহজ সলিউশন হলো একটি ভেক্টর নেয়া। i তম পজিশনে x সংখ্যাটি পেলে ভেক্টরের x তম পজিশনে পুশ করে রাখবে i। তাহলে ইনপুট নেবার পর ভেক্টরগুলোর চেহারা হবে:

[0]->1 2 7  
[1]->0 6 10  
[2]->3 5 11  
[3]->empty  
[4]->8  
[5]->4 9

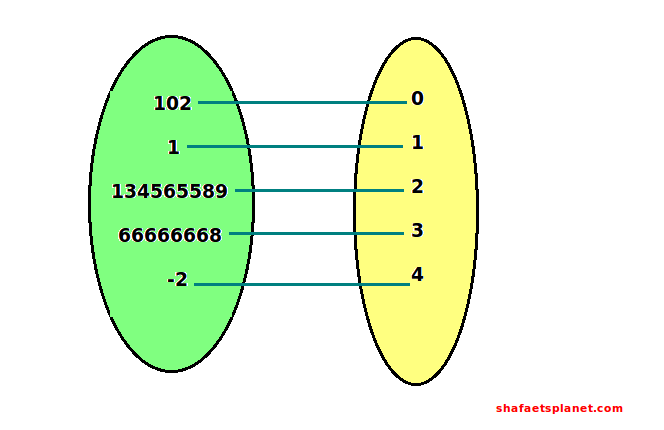
কোন সংখ্যা কোন কোন পজিশনে আছে তুমি পেয়ে গেলে। এখন যদি বলে 2 কোথায় কোথায় আছে তাহলে তুমি ২ নম্বর ইনডেক্সে লুপ চালিয়ে 3,5,11 প্রিন্ট করে দাও। 1001 সাইজের ২-ডি ভেক্টর দিয়েই কাজ হয়ে যাবে।

এখন তোমাকে বলা হলো সংখ্যাগুলো নেগেটিভও হতে পারে এবং সর্বোচ্চ 2^30 পর্যন্ত হতে পারে। এবার কি এই পদ্ধতি কাজ করবে? একটি সংখ্যা -100 বা 2^30 হলে তুমি সেটার পজিশনগুলা কোন ইনডেক্সে রাখবে? অবশ্যই তুমি এতো বড় ভেক্টর ডিক্লেয়ার করতে পারবেনা অথবা নেগেটিভ ইনডেক্স ব্যবহার করতে পারবেনা। ধরো ইনপুট এবার এরকম:

input[]={-102,1,134565589,134565589,-102,66666668,134565589,66666668,-102,1,-2}

একটা ব্যাপার লক্ষ্য করো, সংখ্যা দেয়া হবে সর্বোচ্চ 10^5 বা ১ লাখটা। তাহলে অ্যারেতে মোট ভিন্ন ভিন্ন সংখ্যা হতে পারে কয়টা? অবশ্যই সর্বোচ্চ ১ লাখটা। আমরা প্রতিটি ভিন্ন ভিন্ন সংখ্যাকে ছোটো কিছু সংখ্যার সাথে one-to-one ম্যাপিং করে ফেলবো। উপরের অ্যারেতে ভিন্ন ভিন্ন সংখ্যা গুলো হলো:

-102,1,134565589,66666668,-2

[](http://www.shafaetsplanet.com/planetcoding/?attachment_id=1389)

দুইটা উপায় আছে ম্যাপিং করার। একটা হলো STL এর map ব্যবহার করে। map এর কাজ হলো যেকোনো একটি সংখ্যা, স্ট্রিং বা অবজেক্টকে অন্য একটি মান অ্যাসাইন করা যেটা আমরা উপরের ছবিতে করেছি। আমাদের এই প্রবলেমে কোন সংখ্যা কার থেকে বড় বা ছোটো সেটা দরকার নাই তাই সর্ট না করেই ম্যাপিং করে দিতে পারি। নিচের কোডটা দেখো:

void compress()

{

map<int,int>mymap;

int input[]={-102,1,134565589,134565589,-102,66666668,134565589,66666668,-102,1,-2};

int assign=0,compressed[100],c=0,n=sizeof(input)/sizeof(int); //array size;

for(int i=0;i<n;i++){

int x=input[i];

if(mymap.find(x)==mymap.end()){//x not yet compressed

mymap[x]=assign;

printf("Mapping %d with %d\n",x,assign);

assign++;

}

x=mymap[x];

compressed[ c++ ]=x;

}

printf("Compressed array: ");

for(int i=0;i<n;i++) printf("%d ",compressed[i]);puts("");

}

কোডের আউটপুট হবে:

Mapping -102 with 0  
Mapping 1 with 1  
Mapping 134565589 with 2  
Mapping 66666668 with 3  
Mapping -2 with 4  
Compressed array: 0 1 2 2 0 3 2 3 0 1 4

আমরা ইনপুট অ্যারেতে যখনই একটু নতুন সংখ্যা পাচ্ছি সেটাকে একটা ভ্যালু অ্যাসাইন করে দিচ্ছি এবং আসল ভ্যালুকে ম্যাপের ভ্যালু দিয়ে রিপ্লেস করে আরেকটি অ্যারেতে রেখে দিয়েছি। এরপরে আমরা নতুন অ্যারেটা ব্যবহার করে প্রবলেমটা সলভ করতে পারবো। কুয়েরিতে যদি বলে -2 কোথায় কোথায় আছে বের করো তাহলে তুমি আসলে 4 কোথায় কোথায় আছে বের করবে কারণ mymap[-2]=4।

অনেক সময় কমপ্রেস করার পরেও কোন সংখ্যাটি কার থেকে বড় এটা দরকার হয়। যেমন তোমাকে বলতে পারে যে একটি ভ্যালু নেয়ার পর ছোটো কোনো ভ্যালু আর নিতে পারবেনা। এই তথ্যটা কিভাবে রাখবে উপরের পদ্ধতিতে ১ এর থেকে -২ এ অ্যাসাইন করা ভ্যালু বড় হয়ে গিয়েছে। যদি তুমি সেটা না চাও তাহলে আগে ভিন্ন ভিন্ন ভ্যালুগুলো আরেকটা অ্যারেতে সর্ট করে নাও।

sorted[]={-102,-2,1,66666668,134565589}

এইবার sorted অ্যারেটা ব্যবহার করে ম্যাপিং করে ফেলো। এ ক্ষেত্রে আসলে যে সংখ্যাটি sorted অ্যারেতে যে পজিশনে আছে সেটাই হবে তার ম্যাপ করা ভ্যালু।  
এটা যদি বুঝতে পারো তাহলে নিশ্চয়ই ম্যাপ ছাড়াই বাইনারি সার্চ করে তুমি কমপ্রেস করে ফেলতে পারবে অ্যারেটাকে। ইনপুট অ্যারের উপর লুপ চালাও, প্রতিটি x এর জন্য দেখো sorted অ্যারেতে x এর অবস্থান কোথায়। সেই মানটা তুমি আরেকটা অ্যারেতে রেখে দাও:

input[]={-102,1,134565589,134565589,-102,66666668,134565589,66666668,-102,1,-2}  
compressed\_input[]={0,2,4,4,0,3,4,3,0,2,1}

কুয়েরির সময়ও বাইনারী সার্চ করে কমপ্রেস করার মানটা বের করে কাজ করতে হবে।  
তুমি ম্যাপ বা বাইনারি সার্চ যেভাবে ইচ্ছা কমপ্রেস করতে পারো। প্রতিবার map এক্সেস করতে logn কমপ্লেক্সিটি লাগে যেটা বাইনারী সার্চের সমান। stl এ বিভিন্ন class,ডাইনামিক মেমরির ব্যবহারের জন্য map কিছুটা স্লো, তবে টাইম লিমিট খুব tight না হলে খুব একটা সমস্যা হবেনা। map ব্যবহার করলে কোডিং সহজ হয়।

কোনো কোনো গ্রাফ প্রবলেমে নোডগুলা আর এজগুলো কে string হিসাবে ইনপুট দেয়। যেমন ধরো গ্রাফের ৩টা এজ হলো:

3  
BAN AUS  
AUS SRI  
SRI BAN

map এ string কেও integer দিয়ে ম্যাপিং করা যায়। এই সুবিধাটা ব্যবহার করে প্রতিটি নোডকে একটি ভ্যালু অ্যাসাইন করবো:

map<string,int>mymap;

int edge,assign=0;

cin>>edge;

for(int i=0;i<edge;i++)

{

char s1[100],s2[100];

cin>>s1>>s2;

if(mymap.find(s1)==mymap.end()){

printf("Mapping %s with %d\n",s1,assign);

mymap[s1]=assign++;

}

if(mymap.find(s2)==mymap.end()) {

printf("Mapping %s with %d\n",s2,assign);

mymap[s2]=assign++;

}

int u=mymap[s1];

int v=mymap[s2];

cout<<"Edge: "<<u<<" "<<v<<endl;

}

অনেক সময় বলা হতে পারে নোডগুলোর মান হবে -2^30 থেকে 2^30 পর্যন্ত কিন্তু সর্বোচ্চ নোড হবে ১০০০০ টা, তখন আমরা নোডগুলোকে একই ভাবে ছোটো ভ্যালু দিয়ে ম্যাপিং করে ফেলবো।

2-d grid ও কমপ্রেস করে ফেলা যায় অনেকটা এভাবে। সেটা নিয়ে আরেকদিন আলোচনা করতে চেষ্টা করবো, তবে নিজে একটু চিন্তা করলেই বের করতে পারবে।

প্রবলেম:  
[Drunk](http://www.lightoj.com/volume_showproblem.php?problem=1003) (loj-1003)  
[Babel](http://uva.onlinejudge.org/external/114/11492.html) (uva-11492)  
[A Node Too Far](http://uva.onlinejudge.org/external/3/336.html) (uva-336)

[Emtiaj Hasan](http://www.facebook.com/emtiajium)

একটা স্ট্রিংকে ম্যাপিং করে যে ভ্যালু পাচ্ছি সেটা কোন স্ট্রিংকে ম্যাপিং করেছে সেটা স্টোর করা ছাড়া ( আমি যেভাবে করেছি আরকি!) কোন বিল্ট ইন ফাংশান দিয়ে বের করা যায়?  
মানে আমি যদি CTG কে 1 দিয়ে ম্যাপিং করি তাহলে 1 = ? এটা দেখাতে হবে।

#include<stdio.h>

#include<iostream>

#include<string>

#include<map>

using namespace std;

string arr[105],str,str2;

map< string, int > city;

int E,asg,a,b,i;

int main(){

//freopen("input.txt","r",stdin);

while(scanf("%d",&E)==1){

asg=0;

for(i=0;i<E;i++){

cin>>str>>str2;

if(city.find(str)==city.end()) city[str]=asg++;

if(city.find(str2)==city.end()) city[str2]=asg++;

a=city[str];

b=city[str2];

arr[a]=str;

arr[b]=str2;

}

for(i=0;i<city.size();i++)

cout<<i<<" represents "<<arr[i]<<endl;

puts("");

city.clear();

}

return 0;

}

শাফায়েত

আমার অন্য কোনো উপায় জানা নেই, আমিও এভাবে করি।

Emtiaj Hasan  
প্রথম ছবিটাতে 102 এর আগে মাইনাস সাইনটা দেখা যাচ্ছেনা(পেইন্টিং মিস্টেইক)।

[**ডিরেকশন অ্যারে**](http://www.shafaetsplanet.com/planetcoding/?p=1448)

লেখক: শাফায়েত

অনেক সময়ই এমন প্রবলেম থাকে যেখানে বলা হয় তুমি একটা ২-ডি অ্যারের কোনো এক পজিশনে আছো, সেখান থেকে তুমি উপরে-নিচে-বামে-ডানে যেতে পারবে। অথবা দাবার বোর্ডে একটা ঘোড়া আছে, তাকে ৮টা দিকে মুভ করানো যায়, এখন কোনো একটা পজিশনে শর্টেস্ট পাথে যেতে হবে। বিগিনার কোডাররা এ ধরণের প্রবলেমে ছোট্ট একটা ট্রিকস না জানার কারণে কোডের সাইজ বিশাল বানিয়ে ফেলে।

ডিরেকশন অ্যারের ট্রিকসটা যারা যানেনা এ ধরণের প্রবলেমে তাদের কোড হয় অনেকটা এরকম:

int x=5,y=3,row=5,col=5,nx,ny;

nx=x+1; ny=y;

if(nx>=1 && nx<=row && ny>=1 && ny<=col)

{

DO SOMETHING

}

nx=x-1; ny=y;

if(nx>=1 && nx<=row && ny>=1 && ny<=col)

{

DO SOMETHING

}

nx=x; ny=y+1;

............

............

............

এভাবে বারবার একই লাইন লিখতে লিখতে কোডের চেহারা ভয়াবহ হয়ে যায়, আর কাওকে কোড দেখতে দিলে তারও পাগল হবার অবস্থা হয়! ৮ ডিরেকশনে মুভ করা গেলেতো কথাই নেই। সবথেকে বড় সমস্যা হলো এক জায়গায় চেঞ্জ করলে সবজায়গায় চেঞ্জ করতে হয়।

একটা improvement হতে পারে if(nx>=1 && nx<=row && ny>=1 && ny<=col) এই লাইনের কন্ডিশনটা একটা ম্যাক্রো বানিয়ে ফেলা। যেমন:

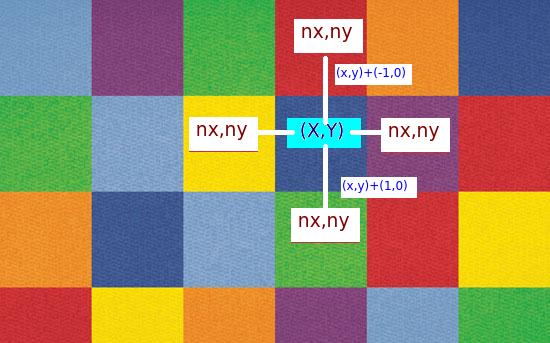
#define valid(nx,ny) nx>=1 && nx<=row && ny>=1 && ny<=col

এখন if(valid) লিখলেই হচ্ছে। এরপরে DO SOMETHING অংশের কাজগুলোও একটা ফাংশন বানিয়ে ফেললে ঝামেলা কিছুটা কমে, এখন খালি ম্যাক্রো বা ফাংশনে চেঞ্জ করলে সব জায়গায় চেঞ্জ হয়ে যাবে। তারপরেও বার বার কন্ডিশন চেক বা ফাংশন কল করতে হচ্ছে আমাদের।

এজন্য আমরা ব্যবহার করবো ডিরেকশন অ্যারে। ৪ দিকে মুভ করা যায় এটার অর্থ হলো:

১. current row এর সাথে ১ যোগ এবং current col এর সাথে ০ যোগ  
২. current row এর সাথে -১ যোগ এবং current col এর সাথে ০ যোগ  
৩. current row এর সাথে ০ যোগ এবং current col এর সাথে ১ যোগ  
৪. current row এর সাথে ০ যোগ এবং current col এর সাথে -১ যোগ

তারমানে x,y পজিশন থেকে উপরে যেতে হলে (x,y)+(1,0) করতে হবে, নিচে যেতে (x,y)+(-1,0) করতে হবে, একই ভাবে ডানে-বামে যেতে যোগ করতে হবে শুধু y এর সাথে।

[](http://www.shafaetsplanet.com/planetcoding/?attachment_id=1459)

আমরা দুটি অ্যারে ডিক্লেয়ার করি এভাবে:

int fx[]={+1,-1,+0,+0};

int fy[]={+0,+0,+1,-1};

fx[] দিয়ে বুঝাচ্ছি row এর সাথে কত যোগ করবো এবং fy দিয়ে বুঝাচ্ছি y এর সাথে কত যোগ করবো। এবার কাজ খুব সহজ হয়ে গেলো:

#define valid(nx,ny) nx>=1 && nx<=row && ny>=1 && ny<=col

int x=5,y=3,row=5,col=5,nx,ny;

for(int k=0;k<4;k++)

{

int nx=x+fx[k]; //Add fx[k] with current row

int ny=y+fy[k]; //Add fy[k] with current col

if(valid(nx,ny)

{

DO SOMETHING;

}

}

তুমি ৮ দিকে যেতে চাইলে অ্যারেটা হবে এরকম:

int fx[]={+0,+0,+1,-1,-1,+1,-1,+1};

int fy[]={-1,+1,+0,+0,+1,+1,-1,-1};

একটু চিন্তা করলেই তুমি দাবার ঘোড়ার মুভের জন্যেও ডিরেকশন অ্যারে লিখতে পারবে। ৩-ডি তেও এটা কাজ করবে, তখন fx[] নামের আরেকটা অ্যারে লাগবে।  
তুমি যদি সম্পূর্ণ কোড চাও তাহলে এই বিএফএস এর কোডটা দেখতে পারো:

//Problem link:http://acm.timus.ru/problem.aspx?space=1&num=1145

#define rep(i,n) for(int i=0; i<(int)n; i++)

#define pb(x) push\_back(x)

#define mem(x,y) memset(x,y,sizeof(x));

#define pii pair<int,int>

#define pmp make\_pair

#define uu first

#define vv second

using namespace std;

#define READ(f) freopen(f, "r", stdin)

#define WRITE(f) freopen(f, "w", stdout)

int fx[]={1,-1,0,0};

int fy[]={0,0,1,-1};

int mr,mc,mx=0;

char w[1000][1000];

int d[1000][1000];

int r,c;

void bfs(int x,int y,int dep)

{

mem(d,63);

d[x][y]=0;

queue<pii>q;

q.push(pii(x,y));

while(!q.empty())

{

pii top=q.front(); q.pop();

if(d[top.uu][top.vv]>mx)

{

mx=d[top.uu][top.vv];

mr=top.uu;

mc=top.vv;

}

rep(k,4)

{

int tx=top.uu+fx[k];

int ty=top.vv+fy[k];

if(tx>=0 and tx<r and ty>=0 and ty<c and w[tx][ty]=='.' and d[top.uu][top.vv]+1<d[tx][ty])

{

d[tx][ty]=d[top.uu][top.vv]+1;

q.push(pii(tx,ty));

}

}

}

}

int main(){

// READ("in");

int sx,sy,cc=0;

cin>>r>>c;

swap(r,c);

rep(i,r)

cin>>w[i];

rep(i,r)

rep(j,c)

if(w[i][j]=='.'){sx=i;sy=j;cc++;}

bfs(sx,sy,0);

mx=0;

bfs(mr,mc,0);

if(cc==1) mx=0;

cout<<mx<<endl;

return 0;

}

এই কোডটা শুধু ডিরেকশন অ্যারের ব্যবহার দেখানোর জন্য, প্রবলেমটার সলিউশন বের করার কাজ তোমার, [ট্রি এর ডায়ামিটার](http://www.shafaetsplanet.com/planetcoding/?p=521) বের করতে বলা হয়েছে প্রবলেমটায়।

হ্যাপি কোডিং!

[আলিম](http://rogrammingzone1.blogspot.com)

ভাইয়া, Hashing নিয়ে একটা Post দিলে ভালো হতো। Uva তে অনেক প্রবলেম আছে যেঁগুলোর tag হোল Hashing… কিন্তু এটা কি জিনিস আমি জানি না। Please, সময় করে একটা Post দিবেন।

শাফায়েত

টপকোডারে Rabin karp এর উপর সুন্দর একটা টিউটোরিয়াল আছে, সেটা পড়লে সহজেই হ্যাশিং বুঝতে পারবে, [লিংক।](http://community.topcoder.com/tc?module=Static&d1=tutorials&d2=stringSearching)

মাঈন উদ্দিন

ভাই , আমি আপনার দেয়া লিংক -এ গিয়েছিলাম কিন্তু ওখানে hashing নিয়ে লেখা এরকম কোনো tutorial খুজেঁ পাই নি ।

শাফায়েত

Rabin-Karp Algorithm অংশটা দেখো। এই অ্যালগোরিদমটা হ্যাশিং দিয়ে কাজ করে, এটা হ্যাশিং এর উপর টিউটোরিয়াল না হলেও কিভাবে হ্যাশিং করা হয় জানতে পারবে।

[**ব্যাকট্র্যাকিং অ্যালগোরিদম: পারমুটেশন জেনারেটর**](http://www.shafaetsplanet.com/planetcoding/?p=1266)

লেখক: শাফায়েত

ব্যাকট্র্যাকিং একধরণের ব্রুটফোর্স অ্যালগোরিদম,ব্রুটফোর্সের মতই এটা সম্ভাব্য সবধরণের বিন্যাস-সমাবেশ থেকে ফলাফল খুজে নিয়ে আসে। যেমন ধরো তোমাকে উত্তরা থেকে শাহবাগ যাবার সবথেকে ছোটো পথ খুজে বের করতে বললো,তুমি ডায়াক্সট্রার দেয়া অ্যালগোকে উড়িয়ে দিয়ে উত্তরা থেকে শাহবাগে যাবার যত পথ আছে সবগুলো খুজে বের করলে এবং তারপর তারমধ্যে থেকে সবথেকে ছোটো কোনটা সেটা বের করলে,এটা হলো ব্যাকট্র্যাকিং। একই সাথে ব্যাকট্র্যাক হলো [ডেপথ ফার্স্ট সার্চ](http://www.shafaetsplanet.com/planetcoding/?p=973) অ্যালগোরিদমের একটি উদাহরণ,তাই [ডিএফএস](http://www.shafaetsplanet.com/planetcoding/?p=973) জানা থাকলে বুঝতে সুবিধা হবে।

সার্চস্পেসের আকার ছোটো হলে এটা খুবই কার্যকর একটা পদ্ধতি,সার্চস্পেস হলো কতটুকু অংশজুড়ে তোমার সলিউশন থাকতে পারে সেইটুকু। যেমন তোমাকে যদি ১০সাইজের দুটি সেট দিয়ে বের করতে বলে ২য় সেট ১মটির subset নাকি তাহলে খুব সহজে তুমি ব্যাকট্র্যাক করে 2^10=1024 টি সেট বের করে সবগুলোর সাথে মিলিয়ে দেখতে পারো,কিন্তু সেটের আকার ১০০ হলে তোমার এভাবে সলিউশন বের করার জন্য এই জীবনকাল যথেষ্ট নয়! যেসব প্রবলেমের পলিনোমিয়াল কোনো সলিউশন নেই অর্থাত [NP-complete](http://en.wikipedia.org/wiki/NP-complete) প্রবলেম সেগুলোকে ব্যাকট্র্যাক করেই সমাধান করতে হয়।

ব্যাকট্র্যাকে যেহেতু প্রতিটি সম্ভাব্য ফলাফল দেখে আসল ফলাফল বের করা হয় তাই সাধারণত কনটেস্টে ব্যাকট্র্যাক প্রবলেমে দুই রকমের verdict আসে: Accepted,time Limit Exceeded। ব্যাকট্র্যাক সলিউশন লেখার আগে তাই ইনপুটের আকার দেখে নেয়া এবং সঠিকভাবে কমপ্লেক্সিটি হিসাব করা খুবই জরুরী।

এখন আমরা দেখবো কিভাবে ব্যাকট্র্যাক করে সম্ভাব্য সবগুলো বিন্যাস-সমাবেশ তৈরি করা যায়। তোমাকে একটি অ্যারে দেয়া আছে:

arr[]={‘a’,'b’,'c’,'d’}

তোমার কাজ হলো অ্যারেটির প্রতিটি পারমুটেশন প্রিন্ট করা(যে কাজটা [next\_permutation](http://www.cplusplus.com/reference/algorithm/next_permutation/) ফাংশন দিয়ে করা যায়)। অর্থাৎ আউটপুট হবে:

abcd  
abdc  
acbd  
….  
….  
dcba  
(মোট ৪!=২৪ লাইন)

একটা সলিউশন হলো ৪টি nested loop চালিয়ে দেয়া। কিন্তু অ্যারের সাইজ ৪ না হয়ে ১০ হলে তখন ১০টি লুপ লিখতে হবে! এরথেকে রিকার্সিভ ব্যাকট্র্যাকিং করে সহজেই আমরা সলিউশন বের করতে পারি। মনে করি আমাদের একটি ভেক্টর/অ্যারে আছে যার নাম result। আমরা প্রতি রিকার্সিভ কল এ arr অ্যারে থেকে একটি করে ক্যারেক্টার result অ্যারেতে ইনসার্ট করবো,এবং রিকার্সিভলি সলিউশন খুজতে চেষ্টা করবো। এরপর ইনসার্ট করা ক্যারেক্টারটি ফেলে দিয়ে অন্য আরেকটি ক্যারেক্টার নিবো তারপর আবার ফলাফল খুজে বের করবো,তার সেটা ফেলে দিয়ে আরেকটি নিবো,এভাবে চলতে থাকবে। নিচের কোডটি দেখো,এরপরে ব্যাখ্যা করছি:

char arr[]={'a','b','c','d'};

vector<char>result;

int taken[20]={0};

void call()

{

if(result.size()==4)

{

for(int i=0;i<4;i++)

printf("%c",result[i]);

puts("");

return;

}

for(int i=0;i<4;i++)

{

if(taken[i]==0)

{

taken[i]=1; result.push\_back(arr[i]);

call();

taken[i]=0; result.pop\_back();

}

}

}

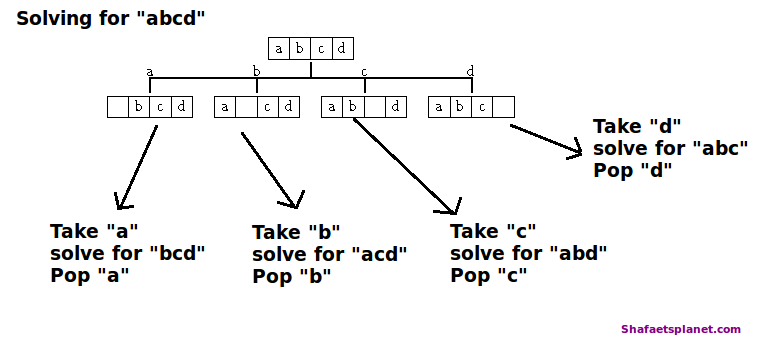
int main()

{

call();

}

১৫ নম্বর লাইনে for লুপটিতে কোনো একটি সংখ্যাকে এর মধ্যে না নেয়া হলে সেটাকে result ভেক্টরে পুশ করে দেয়া হচ্ছে এবং রিকার্সিভলি ফাংশন কল করা হচ্ছে। তাহলে এখন যেটা পুশ করা হলো সেটা পরবর্তী ফাংশনগুলো আর পুশ করবেনা(কারণ taken[i]=১ হয়ে যাবে)। পরবর্তী ফাংশনগুলো যেসব ক্যারেক্টার এখনো নেয়া হয়নি সেগুলো থেকে সমাধান বের করার চেষ্টা করবে। সমাধান বের করা হয়ে গেলে যেটা পুশ করেছিলাম সেটা ফেলে দিবো এবং পরবর্তী ক্যারেক্টার পুশ করে আবার একই কাজ করবো। যখন result ভেক্টরের আকার ৪ হয়ে যাবে তখন বুঝবো আমরা একটি সমাধান পেয়েছি,তখন সেটা সেভ করে বা প্রিন্ট করে ফাংশন থেকে বের হয়ে যাবো। নিজের ছবিটা দেখো:

[](http://www.shafaetsplanet.com/planetcoding/wp-content/uploads/2012/10/backtrack.png)

ছবিটি দেখেই অ্যালগোরিদমটা স্পষ্ট বুঝা যাচ্ছে। ব্যাকট্র্যাকিং করার জেনারেল আইডিয়াটা হলো:

১. প্রতিটি ফাংশন কলে সম্ভাব্য অপশনগুলোর একটি বাছাই করো।  
২. বাকি অপশনগুলো থেকে রিকার্সিভলি সমাধান বের করার চেষ্টা করো।  
৩. বাছাই করা অপশনটি ফেলে দিয়ে অন্য আরেকটি নিয়ে আবার চেষ্টা করো।

আমাদের পারমুটেশন জেনারেট প্রবলেমে:

১.সম্ভাব্য অপশন থেকে বাছাই মানে হলো যেসব ক্যারেক্টার এখনো নেয়া হয়নি সেগুলোর একটি।  
২.বাকি অপশন হলো যেসব ক্যারেক্টার নেয়া হয়নি সেগুলো,এদের জন্য এখন রিকার্সিভলি সলভ করতে হবে।  
৩.যে ক্যারেক্টারটি নিয়েছিলাম সেটা বাদ দিয়ে আরেকটি নিয়ে একই কাজ করবো।

তুমি যেসব প্রবলেম ব্যাকট্র্যাক দিয়ে করতে পারবে তারমধ্যে আছে পারমুটেশন/কম্বিনেশন জেনারেটর,সবগুলো সাবসেট জেনারেটর,ট্রাভেলিং সেলসম্যানসহ বিভিন্ন NPcomplete প্রবলেম,N-queens প্রবলেম ইত্যাদি,তবে সার্চস্পেস অবশ্যই ছোটো হতে হবে। ব্যাকট্র্যাকিং অ্যালগোরিদমের কমপ্লেক্সিটি পলিনোমিয়াল না হয়ে exponential হয়(constant^n আকারের) তাই n এর মান অল্প বৃদ্ধিতেই কমপ্লেক্সিটি অনেক বেড়ে যায়।

উপরের কোডটি বুঝে থাকলে তুমি এখন কোডটাকে কিছুটা চেঞ্জ করতে পারো,যেমন একই সংখ্যা যতবার ইচ্ছা নেয়া গেলে যতগুলো পারমুটেশন তৈরি করা সম্ভব সবগুলো প্রিন্ট করো অথবা অ্যারের ৪টি এলিমেন্ট থেকে যেকোনো ৩টি করে এলিমেন্ট নিলে যতভাবে সাজানো যায় সবগুলো দেখাও। সলভ করার জন্য কতগুলো প্রবলেম:

[Determine The combination](http://uva.onlinejudge.org/external/107/10776.html) (uva - 10776)  
[Prime Ring problem](http://uva.onlinejudge.org/external/5/524.html) (uva - 524)  
[House of santa clause](http://uva.onlinejudge.org/external/2/291.html) (uva - 291)  
[All Walks of length n](http://uva.onlinejudge.org/external/6/677.html) (uva - 677)  
[Following orders](http://uva.onlinejudge.org/external/1/124.html) (uva - 124)

মুকিত

UVA 524 এ WA দিচ্ছে… কোন special কিছু জানা আছে তোমার,যার কারণে WA দিতে পারে ?? আমি কোন error বের করতে পারিনি… :(

মুকিত

৮ অক্টোবর ২০১২

Accepted !!! After 5 WA !!! Frustrated হয়ে পড়ছিলাম !!!

মুকিত

uva 677 এ,  
১) একই নোডের মধ্যে কি connection থাকতে পারে ???  
২) গ্রাফ কি always connected হবে ?

[**ইউভিএ ১০৭০২(ট্রাভেলিং সেলসম্যান)**](http://www.shafaetsplanet.com/planetcoding/?p=571)

লেখক: শাফায়েত

ট্রাভেলিং সেলসম্যান মূলত একটি NP-complete ক্লাসিকাল প্রবলেম। [uva 10702](http://uva.onlinejudge.org/external/107/10702.html) **মূল প্রবলেমের পরিবর্তিত রূপ** যা ডাইনামিক প্রোগ্রামিং এর মাধ্যমে সমাধান করা সম্ভব। একজন ফেরিওয়ালা এক শহর থেকে অন্য শহরে ঘুরে জিনিসপত্র বিক্রি করে। সে কোনো শহরে একাধিক বার যেতে পারে(ক্লাসিক tsp প্রবলেমে একটি নোড একবারের বেশি ভিজিট করা যায়না)। একটি শহর থেকে আরেকটি শহরে গেলে নির্দিষ্ট পরিমাণ লাভ হয়। ফেরিওয়ালা নির্দিষ্ট কিছু শহরে তার যাত্রা শেষ করে এবং যাত্রা শেষ করার আগে নির্দিষ্ট সংখ্যক বার বিভিন্ন শহরে ভ্রমণ করে(inter-city travel)। বলতে হবে সে সর্বোচ্চ কত লাভ করতে পারবে। এই প্রবলেমে গ্রাফটি complete,সব শহর থেকে সব শহরে যাওয়া যায়।

sample ইনপুটে ফেরিওয়ালা সর্বোচ্চ ২বার ভ্রমণ করতে পারবে, ভ্রমণের পথ হবে ১-৩,৩-২,এতে লাভ হবে ৫+২=৭ যা এই গ্রাফের জন্য সর্বোচ্চ।

ডাইনামিক প্রোগ্রামিং এর সাহায্যে সমস্যাটি সমাধান করা যায়। ধরি F(n,d) হলো আমাদের ফাংশন। ধরা যাক আমি এখন n তম শহরে আছি এবং আমি আর d সংখ্যক ভ্রমণ করতে পারব, f(n,d) রিটার্ণ করবে এ অবস্থায় আমার পক্ষে সর্বোচ্চ যত লাভ করা সম্ভব সেটা। কিন্তু সর্বোচ্চ লাভ পাব কি করে? মনে করি n তম শহর থেকে a,b তম শহরে যাওয়া যায়। তাহলে:

f(n,d)=max( w[n][a]+ ‘a তম শহর থেকে d-1 বার ভ্রমন করে গন্তব্য পৌছানোর পর অর্জিত মুনাফা’ , w[n][b]+ ‘b তম শহর থেকে d-1 বার ভ্রমন করে গন্তব্য পৌছানোর পর অর্জিত মুনাফা’ )  
বা,f(n,d)=max( w[n][a]+F(a,d-1) , w[n][b]+F(b,d-1) )

এই ফাংশনটি ভালো ভাবে বুঝলেই সমস্যা সমাধান হয়ে যাবে। এখন আমাদের দরকার একটি base case। যখন d==1 তখন অবশ্যই আমাদেরকে কোনো একটি “গন্তব্য শহর” এ ভ্রমণ করতে হবে কারণ তারপর আর ভ্রমণ করা সম্ভব নয়। যেখানে গেলে সর্বোচ্চ লাভ হবে সেটা রিটার্ন করে দিতে হবে এ ক্ষেত্রে। আমি ফাংশনটি লিখেছি এভাবে:

i64 call(i64 u,i64 d)

{

if(d==0) return 0;

if(!vis[u][d])

{

vis[u][d]=1;

i64 MX=-1\*1e16;

if(d==1)

{

for(int i=1;i&lt;=N;i++)

if(last[i]) MX=max(MX,mat[u][i]);

return dp[u][d]=MX;

}

for(int i=1;i&lt;=N;i++)

{

i64 ret=mat[u][i]+call(i,d-1);

MX=max(ret,MX);

}

return dp[u][d]=MX;

}

return dp[u][d];

}

এ ধরণের আরেকটি প্রবলেম হলো [10681](http://uva.onlinejudge.org/external/106/10681.html),ওটায় গ্রাফটি complete না এবং কন্ডিশনগুলো কিছুটা অন্যরকম।

ক্লাসিক travelling salesman প্রবলেম নিয়ে জানতে দেখো: <http://en.wikipedia.org/wiki/Travelling_salesman_problem> ,এছাড়া IIT এর Prof. G.Srinivasan এর লেকচার দেখতে [ক্লিক করো এখানে](http://www.youtube.com/watch?v=-cLsEHP0qt0&feature=relmfu)।

<http://uva.onlinejudge.org/external/109/10944.html> এই প্রবলেমটি ক্লাসিক travelling salesman প্রবলেমের বেশ কাছাকাছি,নোড কম থাকায় বিটমাস্কিং করে সলভ করা যায়।

**মন্তব্য**

“for(int i=1;i<=N;i++)” এখানে লুপের ভিতরে কন্ডিশনটা কিভাবে চেক করা হচ্ছে বুঝলাম না।

শাফায়েত

লুপের মধ্যে প্রতিবার নতুন একটা শহরে যাচ্ছি , i তম শহরে যাবার কস্ট mat[u][i], এবং সেখানে যাবার পর নতুন স্টেট হবে (i,d-1) । ম্যাক্সিমামটা রিটার্ণ করবো।  
আর d==1 হলে শুধু টেস্ট করতেসি ফাইনাল শহরগুলোতে পৌছানো সম্ভব নাকি এবং কত কম কস্ট এ।

[**সি++ ম্যাপ(map) এ [] অপারেটর ব্যবহারে সতর্কতা**](http://www.shafaetsplanet.com/planetcoding/?p=525)

লেখক: শাফায়েত

সি++ এ ম্যাপ ব্যবহার করলে ম্যাপে কোনো এলিমেন্ট আছে নাকি সেটা আমরা ২ভাবে দেখতে পারি। if(MP[x]==0){} এভাবে অথবা find() ফাংশন ব্যবহার করে। প্রথম পদ্ধতিতে দ্রুত কাজ করা গেলেও এতে সমস্যা হলো [] অপারেটরের মূল কাজ ম্যাপে এলিমেন্ট যোগ করা। তাই চেক করার সময় এলিমেন্টটি ম্যাপে যোগ হয়ে যায়,এতে অনেক মেমরি অপচয় হয়,সময়ও বেশি লাগতে পারে। cplusplus.com এ লেখা আছে:

If x does not match the key of any element in the container, the function inserts a new element with that key and returns a reference to its mapped value. Notice that this always increases the map size by one, even if no mapped value is assigned to the element

নিচের কোডটি দেখুন:

int main()

{

map<int,int>MP;

MP[100]=2;

MP[200]=2;

cout<<MP.size()<<endl;

if(!MP[234]) puts("NOT FOUND");

if(MP[123]==0) puts("NOT FOUND");

cout<<MP.size()<<endl;

return 0;

}

আউটপুট প্রিন্ট করলে দেখতে পাবেন ২৩৪ আর ১২৩ সংখ্যাদুটি ম্যাপে আছে নাকি চেক করার পর ম্যাপের আকার বেড়ে গিয়েছে। সুতরাং কোডটি লেখা উচিত এভাবে:

int main()

{

map<int,int>MP;

MP[100]=2;

MP[200]=2;

cout<<MP.size()<<endl;

if(MP.find(234)==MP.end()) puts("NOT FOUND");

if(MP.find(123)==MP.end()) puts("NOT FOUND");

cout<<MP.size()<<endl;

return 0;

}

find() ফাংশনটি এলিমেন্ট খুজে পেলে একটি ইটারেটর রিটার্ন করে যা এলিমেন্টটিকে পয়েন্ট করে,না পেলে .end() ইটারেটর রিটার্ন করে।

[**কম্বিনেটোরিক্স: অ্যারেঞ্জমেন্ট এবং ডি-রেঞ্জমেন্ট গণনা**](http://www.shafaetsplanet.com/planetcoding/?p=600)

লেখক: শাফায়েত

**[[পুরোনো লেখা নতুন করে লিখে সামনে নিয়ে আনলাম। ইনক্লুশন-এক্সক্লুশনের সাহায্যে ডি-রেঞ্জমেন্ট গণনার পদ্ধতি এবং কিছু ডায়াগ্রাম লেখাটায় যোগ করা হয়েছে]]**

কনটেস্ট প্রোগ্রামিং এর একটা দারুণ ব্যাপার হলো কনটেস্টেন্টদের শুধু ভালো প্রোগ্রামিং জানলেই হয়না, সাথে ভালো গণিতও জানা দরকার হয়। বিশেষ করে কম্বিনেটরিক্স আর প্রোবাবিলিটিতে ভালো ধারণা থাকলে অনেক ধরণের প্রবলেম সলভ করে ফেলা যায়।

৪টি টুপি পাশাপাশি সাজানো আছে, টুপিগুলোকে যথাক্রমে ১,২,৩,৪ সংখ্যাগুলো দিয়ে চিহ্ন দেয়া হয়েছে। এখন টুপিগুলোকে এলোমেলো করে কতভাবে সাজানো যাবে? আমরা কয়েকভাবে সাজিয়ে চেষ্টা করি:

১,২,৩,৪  
১,৩,২,৪  
১,৪,২,৩  
১,৩,৪,২  
…….  
……  
৪,৩,২,১

মোট কতভাবে সাজানো যাবে? কলেজে করে আসা অংক থেকে তুমি সহজেই বলতে পারবে factorial(৪)=২৪ ভাবে সাজানো যায়। এটাকে আমরা একটু প্রোগ্রামারের দৃষ্টিভঙ্গী থেকে দেখি। ৪টা জায়গা বা স্লট আছে, প্রতিটি স্লটে ১টি করে টুপি বসানো যায়। এখন প্রথম স্লটে ১,২,৩ বা ৪ এর যেকোনো একটা বসালে:

১,\_,\_,\_

প্রথম স্লটে টুপি কত ভাবে বসানো যায়? অবশ্যই ৪ ভাবে। এখন ২য় স্লটে কয়ভাবে বসানো যায়? একটা টুপি আমরা বসিয়ে ফেলেছি আগেরটায়, তাই ২য় স্লটে বসাতে পারবো ৪-১=৩ ভাবে। ঠিক এভাবে ৩য় স্লটে ২ভাবে এবং ২য় স্লটে ১ ভাবে। তাহলে মোট উপায় ৪\*৩\*২\*১=২৪টা। ৪টার জায়গায় n টা টুপি থাকলে কি করতে? আমরা প্রোগ্রামার তাই বারবার কষ্ট করে হিসাব না করে ধুম করে একটা ফাংশন লিখে ফেলি। মনে করো ফাংশনটা হলো permutation(n)। n=0 হলে সাজানো যায় ১ ভাবে, তাহলে:

permutation(0)=0

n>0 হলে প্রথম স্লটে বসানো যায় n ভাবে, এরপরে সমস্যাটা ছোটো হয়ে দাড়ায় “n-1 টা টুপি n-1 টা স্লটে কতভাবে বসানো যায়?” অর্থাৎ সমস্যাটা permutation(n-1) হয়ে যায়। সাথে গুণ হবে n কারণ কারেন্ট স্লটে n ভাবে বসিয়েছি। তাহলে লিখতে পারি:

permutation(n)=n\*permutation(n-1)

আশা করি ব্যাপারটা পরিষ্কার। সহজ ব্যাপারটা নিয়ে এত কথা বললাম যাতে রিকার্শনটা পরিষ্কার হয় যেটা কাজে লাগবে ডিরেঞ্জমেন্ট গোণার জন্য।

এখন ধরো ১,২,৩,৪ এই ৪টা টুপির মালিক হলো যথাক্রমে সাকিব, নাসির, তামিম, রহিম। তারা খুবই ভালো বন্ধু বলে ঠিক করলো একজন আরেকজনের টুপি পড়ে ক্রিকেট খেলতে যাবে। কেও তার নিজের টুপি পড়তে পারবেনা, তাহলে বন্ধুত্ব থাকবেনা! এখন কতভাবে তারা টুপি পড়তে পারবে?

গণিতের ভাষায় এর নাম ডিরেঞ্জমেন্ট, এমন কয়টি পারমুটেশন আছে যেখানে কেও তার নিজের জায়গায় নেই।

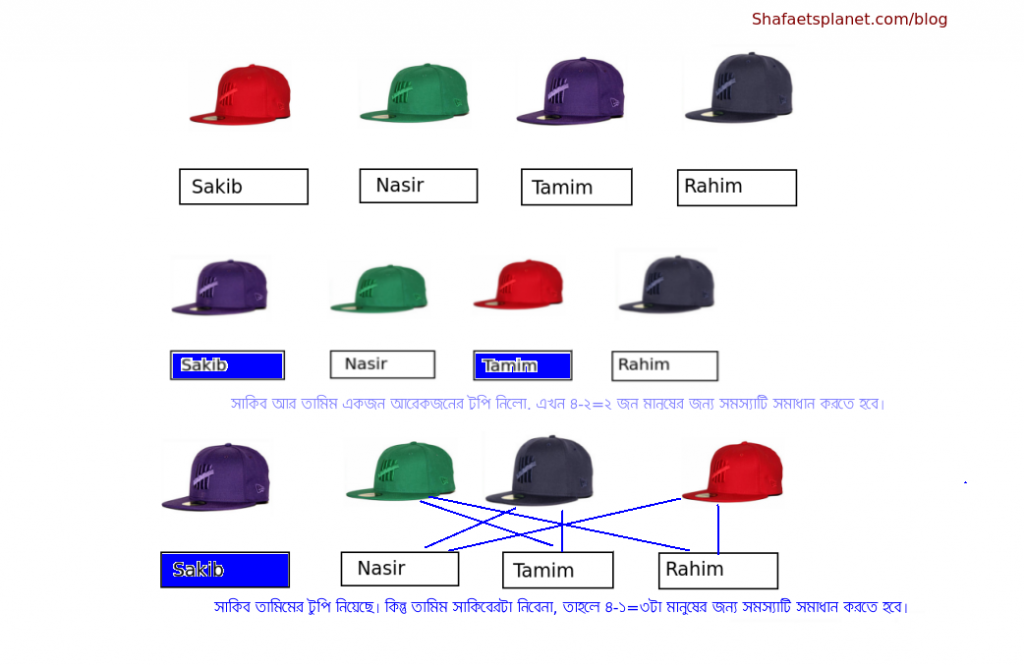
১,৩,২,৪ ডি-রেঞ্জমেন্ট নয় কারন সাকিব আর রহিম তাদের নিজ নিজ টুপিই পড়ে আছে(১ ও ৪ নম্বর) ! ২,১,৪,৩ একটি ডি-রেঞ্জমেন্ট, সবাই তার বন্ধুর টুপি পড়েছে।

আমরা একটা ফাংশন বানাবো d(n) যেটা n টা টুপি কতভাবে সাজানো যায় যাতে কেও তার নিজের টুপি না পায় সেটা বের করে দেয়।

প্রথম মানুষ সাকিবের কাছে ৪-১=৩টা চয়েস আছে, সে ১ নম্বর বাদে যেকোনো টুপি নিতে পারে। মনে করলাম সে তামিমের টুপি নিলো। এখন ২টা ঘটনা ঘটতে পারে:

১. পরের বার তামিম নিলো সাকিবের টুপি। এখন ৪-২=২ জন মানুষ বাকি, টুপিও বাকি ঠিক ৪-২=২ টা।

২. পরের বার তামিম সাকিব ছাড়া অন্য কারো টুপি নিলো। এখন মানুষ বাকি ৪-১=৩ জন। তামিম যেহেতু সাকিবের টুপি নিচ্ছেনা তাই ওটাকেই তার নিষিদ্ধ টুপি ধরতে হবে, আর বাকি সবার কাছে নিষিদ্ধ টুপি হলো তার নিজের টুপিটা। তাহলে এখন ৪-১=৩ জন মানুষের জন্য ৪-১=৩ টা করে চয়েস আছে। লক্ষ্য করো

[](http://www.shafaetsplanet.com/planetcoding/?attachment_id=1532)

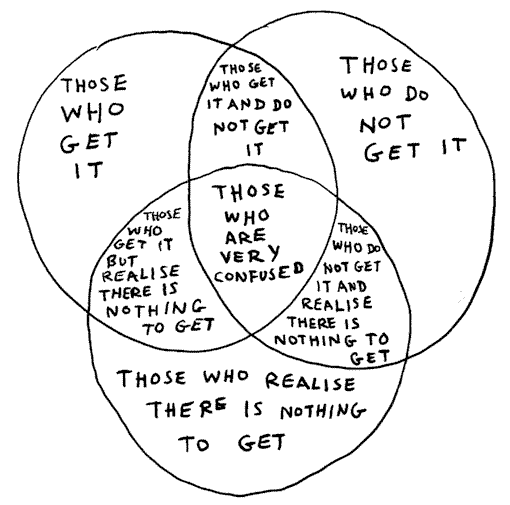
দুই ক্ষেত্রেই মানুষ আর টুপির সংখ্যা সমান থাকছে। ৪ এর জায়গায় n ধরে ২টা কন্ডিশন মিলিয়ে সহজেই রিকার্সিভ রিলেশনটা লিখতে পারি:

d(n)=(n-1)\* ( d(n-1)+d(n-2) )  
বা d(n)=(n-1) ( d(n-1)+d(n-2) )  
বেস কেস: d(1)=0,d(2)=1

এই রিকার্সনটা কোড করার সময় মাথায় রাখতে হবে যে একই ফাংশন অনেকবার কল হচ্ছে,তাই ডিপি টেবিলে মানগুলো সেভ করে রাখতে হবে। তুমি ডাইনামিক প্রোগ্রামিং নিয়ে পড়ালেখা করতে পারো এ সম্পর্কে জানতে।

এবার আরেকটা মজার উপায়ে প্রবলেমটা সলভ করি। nCr এর সাথে তোমরা পরিচিত, n টা জিনিস থেকে r টি জিনিস কতভাবে নেয়া যায় সেটাই প্রকাশ করে nCr। এটাকে সুবিধার আমরা ফাংশন আকারে অর্থাৎ ncr(n,r) এভাবে লিখবো। n টা টুপিকে মোট সাজানো যায় n! উপায়। এর মধ্যে যেসব পারমুটেশনে **অন্তত একটি টুপি** নিজের জায়গায় আছে তাদের বাদ দিলে ডিরেঞ্জমেন্ট পাওয়া যায়। n টি টুপি থেকে ১ টি টুপি নেয়া যায় ncr(n,1) উপায়ে, ১টি টুপিকে নিজের জায়গায় রেখে বাকি n-1 টা টুপিকে সাজানো যায় (n-1)! উপায়ে। তাহলে n!-ncr(n,1)\*(n-1)! বের করলেই ডিরেঞ্জমেন্ট বের হয়ে যাচ্ছেনা? কারণ আমরা মোট উপায় থেকে যেসব পারমুটেশনকে **অন্তত ১ জন** নিজের জায়গায় আছে তাদের বাদ দিচ্ছি। ncr(n,1) দিয়ে গুণ দিচ্ছি কারণ প্রতিবার ১জন কে ফিক্সড করে n-1 জনকে পারমুটেশন করতেসি।

কিন্তু এখানে একটা বড় সমস্যা আছে। ধরো তুমি তামিমের টুপিকে তামিমের কাছেই রেখে বাকি টুপিগুলো কয়ভাবে সাজানো যায় বের করলে। আবার নতুন করে সাকিবেরটা সাকিবের কাছে রেখে বাকিগুলো কয়ভাবে সাজানো যায় বের করলে। ভালোমত চিন্তা করে দেখ যেসব পারমুটেশনে সাকিবেরটা সাকিবের কাছে আছে আর তামিমেরটা তামিমের কাছে আছে সেগুলো কি ২বার গণনা করা হয়ে গেলো না? ncr(n,1)\*(n-1)! এ এই কারণে কিছু পারমুটেশন একাধিক বার ক্যালকুলেট করা হয়ে যাবে। সেগুলো আমরা কিভাবে বাদ দিবো? আমরা ১টা সংখ্যা ফিক্সড করে যখন গুনেছি তখন যেসব পারমুটেশনে ২টা সংখ্যা ফিক্সড সেগুলো একাধিক বার গুণে ফেলেছি, সেগুলো আমরা বাদ দিয়ে দেই। ncr(n,1)\*(n-1)! থেকে বাদ দিয়ে দিবো ncr(n,2)\*(n-2)! । একটু চিন্তা করলে বুঝতে পারবে এখানেও সমস্যা আছে, যেখানে ৩টা ফিক্সড সেগুলোকেও আমরা বাদ দিয়ে দিচ্ছি!! তাহলে সেটা আবার যোগ করে দাও। মাথা গুলিয়ে গেলে ভ্যান ডায়াগ্রামের কথা চিন্তা করো:



ভ্যান ডায়াগ্রামে ৩টা অংশের কমন এরিয়া বের করতে আমরা সবগুলো অংশ যোগ করি, তারপর যেসব অংশ দুটি বুত্তে আছে সেগুলো বাদ দেই, যেগুলো ৩টি বৃত্তে আছে সেগুলো আবার যোগ করে দেই, বৃত্ত আরো বেশি থাকলে এভাবে যোগ বিয়োগ চলতেই থাকে। দুটি সেট A,B হলে |A∪B|=|A|+|B|-|A∩B|। ঠিক এই কাজটি করবো এখানে। আমাদের ফর্মূলা হবে:  
**n!-ncr(n,1)\*(n-1)!+ ncr(n,2)\*(n-2)!-..…+ (-1)^k\*ncr(n,k)\*(n-k)!+….+(-1)n**

আমরা একবার যোগ করছি, একবার বিয়োগ করছি, এভাবে অপ্রয়োজনীয় অংশ বাদ দিয়ে ফলাফল পেয়ে যাচ্ছি। এ জিনিসটারই রাশভারী নাম হলো ইনক্লুশন-এক্সক্লুশন প্রিন্সিপাল।

আজ এ পর্যন্তই। সলভ করার জন্য প্রবলেম:  
[www.topcoder.com/stat?c=problem\_statement&pm=2013](http://www.shafaetsplanet.com/planetcoding/www.topcoder.com/stat?c=problem_statement&pm=2013)  
<http://uva.onlinejudge.org/external/112/11282.html>  
<http://www.lightoj.com/volume_showproblem.php?problem=1095>

**মন্তব্য**

হাসান

ভাইয়া, কয়েকটা প্রবলেমের আইডি বলেন সলভ করার জন্য।

শাফায়েত

<http://uva.onlinejudge.org/external/112/11282.html>  
<http://www.lightoj.com/volume_showproblem.php?problem=1095>

মুন্না

@ হাসান  
এই প্রব্লেমটাও derangement নিয়ে <http://uva.onlinejudge.org/external/120/12024.html>

[**ভাগশেষের গণিত(মডুলার অ্যারিথমেটিক + বিগ মড)**](http://www.shafaetsplanet.com/planetcoding/?p=936)

লেখক: শাফায়েত

-১৭ কে ৫ দিয়ে ভাগ করলে ভাগশেষ কত হয়? ২^১০০০ কে ১৭ দিয়ে ভাগ করলে ভাগশেষ কত হয় সেটা কি তুমি integer overflow এড়িয়ে নির্ণয় করতে পারবে? O(n) এ পারলে O(log n) কমপ্লেক্সিটিতে পারবে? যদি কোনো একটি উত্তর “না” হয় তাহলে এই পোস্ট তোমার জন্য। তবে তুমি যদি মডুলার ইনভার্স বা এডভান্সড কিছু শিখতে পোস্টটি খুলো তাহলে তোমাকে আপাতত হতাশ করতে হচ্ছে।

সি/জাভা সহ বেশিভাগ প্রোগ্রামিং ল্যাংগুয়েজে এ % কে ভাগশেষ অপারেটর ধরা হয়। x কে m দিয়ে ভাগকরে ভাগশেষ বের করার অর্থ x%m এর মান বের করা অথবা আমরা বলতে পারি x কে m দিয়ে mod করা। “determine answer modulo 1000″ এ কথাটির অর্থ হলো উত্তরকে ১০০০ দিয়ে mod করে তারপর আউটপুট দিতে হবে।

একটি সমস্যা দিয়ে শুরু করি। তোমার ১০০টি বই আছে,তুমি কয়ভাবে বইগুলো সাজাতে পারবে? খুব সহজ,১০০! (১০০ ফ্যাক্টরিয়াল) ভাবে সাজাতে পারবে। ১০০! ১৫৮ ডিজিটের বিশাল একটি সংখ্যা। তাই আমি তোমাকে প্রবলেমটা সহজ করে দিলাম,ধরো তুমি x উপায়ে বইগুলো সাজাতে পারবে,তাহলে তোমাকে x%97 কত সেটা বলতে হবে। অর্থাত ১০০! বের করে ৯৭ দিয়ে ভাগ করে ভাগশেষটা বের করাই তোমার সমস্যা। (Determine 100 factorial modulo 97)

এটা কিভাবে করবে? ১০০! এর মান তুমি বের করতে পারবেনা ৬৪বিট আনসাইনড ইন্টিজার দিয়েও, এরা ২^৬৪-১ পর্যন্ত সংখ্যা নিয়ে কাজ করতে পারে, তাই ওভারফ্লো হবে। কিন্তু আমরা জানি আমাদের উত্তর কখনোই 97 এর বড় হবেনা কারণ **কোনো সংখ্যাকে m দিয়ে mod করা হলে সংখ্যাটি m এর থেকে বড় হতে পারবেনা**।

আমরা এ ধরণের সমস্যা সমাধান করতে সাহায্য নিবো দুটি সুত্রের:

**(a+b)%m=((a%m)+(b%m))%m**  
**(a\*b)%m=((a%m)\*(b%m))%m**

n সংখ্যক নম্বর a1,a2…an এর জন্য সুত্র দুটি ব্যবহার করতে পারবে।

উপরের সমস্যাটিতে ২য় সুত্রটি লাগবে। তোমার বের করা দরকার ১০০! %৯৭ অর্থাত:  
(১০০\*৯৯\*৯৮\*………..\*১)%৯৭  
তুমি যেটা করবে সেটা হলো গুণ করার সময় ২য় সুত্রের মত করে mod করতে থাকবে,তাহলে কোনো সময়ই overflow ঘটবেনা কারণ mod করলে প্রতি স্টেপে সংখ্যাটি ছোটো হয়ে যাচ্ছে। এটার কোড হতে পারে এরকম:

int fact=1;

for(int i=1;i<=100;i++)

{

fact=((fact%97)\*(i%97))%97;

}

printf("%d\n",fact);

এটার আউটপুট আসবে ০। অর্থাত ১০০! % ৯৭ =০। একটু খেয়াল করলেই বুঝবে এখানে আমরা ২য় সুত্রটি প্রয়োগ করেছি ২টি করে সংখ্যা নিয়ে।

সুত্র দুটি কেনো কাজ করে সেটা জানা দরকার। আমি ১ম সুত্রটির প্রমাণ দেখাচ্ছি,২য়টিও একইভাবে করা যায়। প্রমানটি আমার নিজের মত করে করা।

ধরি (x+y)%৫ এর মান আমাদের বের করতে হবে। এখন যদি x%5=c1 আর y%5 = c2 হয়,তাহলে x কে আমরা লিখতে পারি 5n1+c1 এবং y কে লিখতে পারি 5n2+c2 যেখানে n1 আর n2 দুটি ইন্টিজার। এটা একদম বেসিক রুল,আশা করে বুঝতে সমস্যা হচ্ছেনা। এখন:

(x+y)%5  
= (5n1+c1+5n2+c2)%5  
=(5n1+5n2+c1+c2)%5 ——(১)

এখানে 5n1+5n2 অবশ্যই 5 এর মাল্টিপল,তাই আমরা লিখতে পারি  
5n1+5n2=5N যেখানে N=n1+n2  
এবং c1+c2=C  
তাহলে (১) থেকে পাচ্ছি:  
(5N+C)%5  
এখন পরিস্কার বোঝা যাচ্ছে যে উত্তর হলো C%5। C কে আবার mod করতে হলো কারণ c1+c2 এর মান 5 এর থেকে বড় হতেই পারে। এখন

((x%5)+(y%5))%5——–(২)  
=((5n1+c1)%5)+((5n2+c2)%5))%5  
(5n1+c1)%5=c1  
(5n2+c2)%5=c2

তাহলে ২ কে লিখতে পারি:  
(c1+c2)%5 = C%5

তাহলে ১ম সুত্রটি প্রমাণিত হলো। তারমান যোগ করে mod করা আর আগে mod করে তারপর যোগ করে আবার mod করা একই কথা। সুবিধা হলে সংখ্যাটি কোনো স্টেপেই বেশি বড় হতে পারেনা। গুণের ক্ষেত্রেই একই সুত্র প্রযোজ্য।

নেগেটিভ সংখ্যার mod নিয়ে একটু আলাদা ভাবে কাজ করতে হয়। সি তে -17 % 5 এর মান দেখায় -২। কিন্তু সচরাচর আমরা ভাগশেষের যে সংজ্ঞা ব্যবহার করি তাতে x%m = p হলে গাণিতিকভাবে

m এর সবথেকে বড় থেকে বড় মাল্টিপল যেটা x এর থেকে ছোট সেই সংখ্যাটিকে x থেকে বিয়োগ করলে যে সংখ্যাটি পাওয়া যায় সেটাই p।

যেমন 23 % 5 এর ক্ষেত্রে ৫\*৪=২০ হলো ৫ এর সবথেকে বড় মাল্টিপল যেটা ২৩ এর থেকে ছোট,তাই 23 % 5=23-(5\*4)=৩। -17 % 5 এর ক্ষেত্র খেয়াল করো -20 হলো ৫ এর সবথেকে বড় মাল্টিপল যেটে -১৭ থেকে ছোট,তাই উত্তর হবে ৩।  
এই কেসটা handle করা একটি উপায় হলো নেগেটিভ সংখ্যাটিকে একটি 5 এর মাল্টিপল এর সাথে যোগ করা যেন সংখ্যাটি ০ থেকে বড় হয়ে যায়,তারপরে mod করা। যেমন:

-17%5  
=(-17+100)%5  
=83%5  
=3

এটা উপরের সুত্রের প্রমাণের মত করেই কাজ করে,একটু গুতালেই প্রমাণ করতে পারবে। **নেগেটিভ সংখ্যার mod নিয়ে কনটেস্টে সবসময় সতর্ক থাকবে,এটা wrong answer খাওয়ার একটা বড় কারণ হতে পারে।**

এবার আসি সুপরিচিত big mod সমস্যায়। সমস্যাটি হলো তোমাকে (a^b)%m এর মান বের করতে হবে,a,b,m তোমাকে বলে দেয়া হবে,সবগুলোর range 2^31 পর্যন্ত হতে পারে। ১০০! % ৯৭ বের করার মত করে সহজেই তুমি overflow না খেয়ে মানটি বের করতে পারবে,সমস্যা হলো তুমি লুপ চালিয়ে একটি একটি গুণ করে (2^(2000000000))%101 বের করতে চাইলে উত্তর পেতে পেতে সম্ভবত নাস্তা শেষ করে আসতে পারবে। আমরা চাইলে O(lgn) এ এটা করতে পারি।  
লক্ষ করো

2^100  
=(2^50)^2  
এবং  
(2^50)  
=(2^25)^2

এখন বলো 2^50 বের করতে কি 2^26,2^27 ইত্যদি বের করার দরকার আছে নাকি 2^25 পর্যন্ত বের করে square করে দিলেই হচ্ছে? আবার 2^25 পর্যন্ত আসতে (2^12)^2 পর্যন্ত বের করে square করে সাথে ২ গুণ করে দিলেই যথেষ্ট,অতিরিক্ত ২ গুণ করছি সংখ্যাটি বিজোড় সে কারণে। প্রতি স্টেপে গুণ করার সময় mod করতে থাকবে যাতে overflow না হয়। recursion ব্যবহার করে কোডটি লেখা জলের মত সোজা:

#define i64 long long

i64 M;

i64 F(i64 N,i64 P)

{

if(P==0) return 1;

if(P%2==0)

{

i64 ret=F(N,P/2);

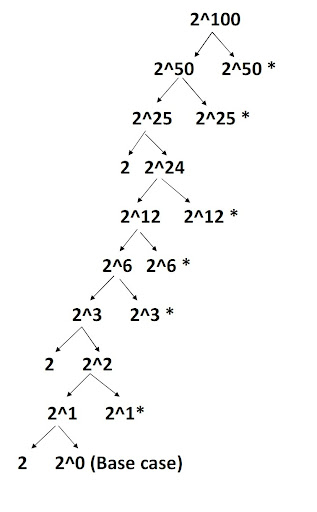
return ((ret%M)\*(ret%M))%M;

}

else return ((N%M)\*(F(N,P-1)%M))%M;

}

মন্তব্য অংশে “হাসান” একটি বিগ মডের সুন্দর রিকার্শন-ট্রি এর ছবির লিংক দিয়েছে, ছবিটা এরকম:



মডুলার অ্যারিথমেটিক ব্যবহার করে বিশাল আকারের ফলাফল কে আমরা ছোট করে আনতে পারি ফলাফলে বিভিন্ন characteristic নষ্ট না করে,তাই এটা গণিতে খুব গুরুত্বপূর্ণ। প্রোগ্রামিং কনটেস্টে প্রায়ই বিভিন্ন প্রবলেমে মডুলার অ্যারিথমেটিক প্রয়োজন পড়বে,বিশেষ করে counting আর combinatorics এ যেখানে ফলাফল অনেক বড় হতে পারে,ফ্যাক্টরিয়াল নিয়ে কাজ করতে হতে পারে।

**ভাগ করার সময় গুণ,আর যোগের মত সুত্র দুটি কাজ করেনা**,এটার জন্য তোমাকে extended euclid আর modular inverse জানতে হবে।

সিপিউর জন্য mod খুব costly একটা অপারেশন। **যোগ,গুণের থেকে mod করতে অনেক বেশি সময় লাগে।** অপ্রয়োজনে mod ব্যবহার করলে কোড time limit exceed করতে পারে,তাই overflow হবার আশংকা না থাকলে সব জায়গায় mod করা দরকার নেই। আমার একটি কোড ৩সেকেন্ডে time limit exceed হবার পর খালি কিছু mod সরিয়ে ১.৩ সেকেন্ড নামিয়ে এনেছি।

এখন চিন্তা করার জন্য একটি প্রবলেম। ধরো তোমাকে একটি অনেক বড় সংখ্যা(bigint) দিয়ে সেটাকে ২^৩১ এর ছোট একটি সংখ্যা দিয়ে mod করতে বলা হলো। O(length\_of\_bigint) কমপ্লেক্সিটিতে কিভাবে করবে?  
সাহায্য:

২৩=(০\*১০+২)\*১০ + ৩  
১২৩৯=(((০\*১০+১)\*১০ + ২)\*১০ +৩)\*১০+৯

প্র্যাকটিসের জন্য প্রবলেম:  
<http://uva.onlinejudge.org/external/3/374.html>  
<http://uva.onlinejudge.org/external/101/10127.html>

**মন্তব্য**

Raihan

i64 দিয়ে কি বুঝিয়েছেন? long long int ব্যবহার করবো?

শাফায়েত

i64 দিয়ে long long বুঝানো হয়েছে।

[শিখর রায়](http://shikhorroy.wordpress.com)

২১ ফেব্রুয়ারি ২০১৩

ভাইয়া ক্যালকুলেটর ইউচ করলে তো -১৭%৫= ২ আসে। মানে আমি করেছি এভাবে -১৭/৫ = -৩ সমস্ত ২ বাই ৫ । কিন্তু আপনি দ্যাখালেন -১৭%৫= ৩ । আমি আসলে এই ব্যাপারটা নিয়ে কনফিউসড । বিভাজ্যতার নিয়ম অনুযায়ী তো আপনি ঠিক আছেন, তাহলে ক্যালকুলেটর কি ভুল? নাকি দুটই ঠিক আছে, আমারি বুঝতে কোথাও প্রব্লেম হচ্ছে !

শাফায়েত

আমিতো আমার উবুন্টুর ক্যালকুলেটরে “−17 mod 5″ লিখে ৩ পাচ্ছি।

[Shikhor Kumer Roy](http://www.facebook.com/shikhorroy.cse12)

ভাইয়া এমনিতে ক্যালকুলেটর চেপে ভাগ করেন -১৭/৫ = -৩.৪ । আর -৩.৪ মানে তো “-৩ সমস্ত ২ বাই ৫” । অর্থাৎ অবশেষ ২ । আমার বোঝায় কি কোথাও সমস্যা আছে?

শাফায়েত

ব্যাপারটা আসলে ঠিক সেরকম না। ধরো একটা সংখ্যা x । n এর যে মাল্টিপলটা x এর সবথেকে কাছাকাছি পৌছায় কিন্তু x এর সমান বা ছোট সেই মাল্টিপল থেকে x এর ডিফারেন্সটাই ভাগশেষ। ৫ এর কোন মাল্টিপল -১৭ এর সবথেকে কাছাকাছি? অবশ্যই -২০। -১৫ না কারণ এটা -১৭ এর থেকে বড়। -২০ আর -১৭ এর পার্থক্য ৩ তাই ভাগশেষ ৩।

[Shikhor Kumer Roy](http://www.facebook.com/shikhorroy.cse12)

তাহলে ব্যাপারটা দাড়ালো,  
যদি বলা হয় -১৭/৫ এর মান কত? উত্তর হবে -৩.৪ [মানে এখানেঃ ভাজ্য, ১৭ = -(৫X৩+২)]  
আর -১৭/৫ এর ক্ষেত্রে ভাগশেষ কত, তাহলে উত্তর হবে ৩ [মানে এখানেঃ ভাজ্য, ১৭ = ৫X(-৪)+৩]  
ব্যাপারটা কেমন জানি সাংঘর্ষিক হয়ে গেলো না ভাইয়া?

[Emtiaj Hasan](http://www.facebook.com/emtiajium)

তুমি ফেইসবুকের মাধ্যমে মাসুম বিল্লাল যে কমেন্টটা দিয়েছে সেটা খেয়াল কর।  
(ভাইয়া আসলে গাণিতিকভাবে -৯%১২=-৯ অথবা ৩। কোনটাই ভুল না। তবে কম্পিউটারের বুদ্ধি নাই যে সে এইটাকে ৩ হিসেবে বলবে :)আসলে ব্যাপারটা হইতেছে -৯ কে এইভাবে দেখেন, ১২ থেকে ৩ সরে আসছে।.)  
আর দেখ এটা–  
মনে কর তোমার কাছে এমন একটা ঘড়ি আছে যেটাতে 12 ঘন্টার জায়গায় মাত্র 5 টা ঘন্টা আছে। এদের মার্ক দেয়া 0, 1, 2, 3, 4. জিরো থেকে শুরু করে ঘড়ির কাটাঁর দিকে ঘুরে তুমি যদি বারো ঘর এগোও তাহলে তুমি 2 এ পৌঁছাবা।

গাণিতিকভাবে আসলেই ব্যপারটা হয়-

12 Ξ 2 (mod 5).

এখন তুমি যদি ঘড়ির কাটাঁর বিপরীত দিকে 12 বার ঘুরে আসো তবে তুমি 3 এ এসে পৌঁছাবা।

- 12 Ξ 3 (mod 5).  
@Shikhor Kumar Roy