快速幂：

2019-11-15 update:

typedef long long int i64;

i64 quickmod(i64 a, i64 b, i64 c) {

i64 ans = 1;

while(b) {

if(b & 1) {

ans = (ans \* a) % c;

}

a = (a \* a) % c;

b >>= 1;

}

return ans;

}

i64 quickmul(i64 a, i64 b, i64 c) { //防溢出

i64 ans = 0;

while(b) {

if(b & 1) {

ans = (ans + a) % c;

}

a = (a + a) % c;

b >>= 1;

}

return ans;

}

素数打表：

n是上限 ， num是下标（从1开始） 0 2 3 5...

bool visit[10100000];

int prime[10000000];

void init\_prim()

{

memset(visit, true, sizeof(visit));

int num = 0;

for (int i = 2; i <= n; ++i)

{

if (visit[i] == true)

{

num++;

prime[num] = i;

}

for (int j = 1; ((j <= num) && (i \* prime[j] <= n)); ++j)

{

visit[i \* prime[j]] = false;

if (i % prime[j] == 0) break;

}

}

}

大数相加：

string sum(string s1,string s2){//大数相加模板

if(s1.length() < s2.length()){string temp = s1;s1 = s2;s2 = temp;}

int i,j;

for(i = s1.length()-1,j = s2.length() - 1;i >= 0;i--,j--){

s1[i] = char(s1[i]+(j >= 0?s2[j]-'0':0));

if(s1[i] - '0' >= 10){

s1[i] = char((s1[i] - '0') % 10 + '0');

if(i) s1[i-1]++;

else s1 = '1' + s1;

}

}

return s1;

}

string form[...(你要打的表的大小)];

因为是字符串 所以输出只能用cout << form[n] << endl;

若要三个数据a,b,c相加 则可用sum(sum(a,b),c);来求 以此类推..

KM匹配：

#define maxn 305

int n , m, con[maxn][maxn], num\_a[maxn], num\_b[maxn], match[maxn], slack[maxn], book\_a[maxn], book\_b[maxn];

bool dfs(int x)

{

book\_a[x] = 1;

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

if (book\_b[i]) continue;

int gap = num\_a[x] + num\_b[i] - con[x][i];

if (!gap) {

book\_b[i] = 1;

if (match[i] == -1 || dfs(match[i])) {

match[i] = x;

return true;

}

} else {

slack[i] = min(slack[i], gap);

}

}

return false;

}

int KM()

{

ms(match, -1); ms(num\_b, 0);

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

num\_a[i] = con[i][0];

for (int j = 1; j < m; ++j) {

num\_a[i] = max(num\_a[i], con[i][j]);

}

}

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

fill(slack, slack + n, INF);

while (1) {

ms(book\_a, 0); ms(book\_b, 0);

if (dfs(i)) break;

int d = INF;

for (int j = 0; j < m; ++j)

if (!book\_b[j]) d = min(d, slack[j]);

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

if (book\_a[j]) num\_a[j] -= d;

}

for (int j = 0; j < m; ++j) {

if (book\_b[j]) num\_b[j] += d;

else slack[j] -= d;

}

}

}

int res = 0;

for (int i = 0; i < m; ++i) res += con[match[i]][i];

return res;

}

int main()

{

ios::sync\_with\_stdio(false);

cin.tie(0);

cout.tie(0);

while (cin >> n) {

m = n;

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

for (int j = 0; j < m; ++j)

cin >> con[i][j];

int ans = KM();

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

}

return 0;

}

2

100 10

15 23

-》 123

矩阵快速幂：

int x, y, n;

struct mat {

ll m[5][5];

}unit;

mat operator \* (mat a, mat b) {

mat res;

for (int i = 1; i <= 2; ++i) {

for (int j = 1; j <= 2; ++j) {

ll x = 0;

for (int k = 1; k <= 2; ++k) {

x += mod(mod(a.m[i][k]) \* mod(b.m[k][j]));

x = mod(x);

}

res.m[i][j] = x;

}

}

return res;

}

void init\_unit() {

memset(unit.m, 0, sizeof(unit.m));

for (int i = 1; i <= 2; ++i) {

unit.m[i][i] = 1;

}

}

mat pow\_mat(mat a, ll n) {

mat res = unit;

while (n) {

if (n & 1) res = res \* a;

a = a \* a;

n >>= 1;

}

return res;

}

int main()

{

ios::sync\_with\_stdio(false);

cin.tie(0);

cout.tie(0);

init\_unit();

cin >> x >> y >> n;

mat a, b;

b.m[1][1] = 1; b.m[1][2] = -1; b.m[2][1] = 1; b.m[2][2] = 0;

b = pow\_mat(b, n-1);

a.m[1][1] = x; a.m[1][2] = y - x;

mat res;

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

for (int j = 1; j <= 2; ++j) {

ll x = 0;

for (int k = 1; k <= 2; ++k) {

x += mod(mod(a.m[i][k]) \* mod(b.m[k][j]));

x = mod(x);

}

res.m[i][j] = x;

}

}

if (res.m[1][1] < 0) {

res.m[1][1] = mod(res.m[1][1] + 1e9 + 7);

}

cout << res.m[1][1] << endl;

return 0;

}

快读写模板：(ll or int)

ll R(){

ll ans=0,f=1;char c=getchar();

for(;c<'0'||c>'9';c=getchar()) if (c=='-') f=-1;

for(;c>='0'&&c<='9';c=getchar()) ans=ans\*10+c-'0';

return ans\*f;

}

inline void write(int x)

{

if(x<0) {

putchar('-');

x = -x;

}

if(x>9) write(x / 10);

putchar(x % 10 + '0');

}

gcd：

unsigned long gcd(unsigned long a, unsigned long b) {

unsigned long r = a | b;

if(!a || !b)

return r;

r &= -r;

while(!(b & r))

b >>= 1;

if(b == r)

return r;

for(;;) {

while (!(a & r))

a >>= 1;

if(a == r)

return r;

if(a == b)

return a;

if(a < b)

swap(a, b);

a -= b;

a >>= 1;

if(a & r)

a += b;

a >>= 1;

}

}

扩展欧几里得：

HDU 6608

#include <bits/stdc++.h>

using namespace std;

\_\_int128 \_a, \_b, \_m;

long long mul (long long a, long long b, long long m) {

\_a = a, \_b = b, \_m = m;

return (long long)(\_a \* \_b % \_m);

}

long long inv (long long a, long long b, long long m) {

long long ret = 1;

while (b) {

if (b & 1) ret = mul(ret, a, m);

a = mul(a, a, m);

b >>= 1;

}

return ret;

}

inline bool check (long long n) {

for (long long i = 2; i \* i <= n; i++) {

if (n % i == 0) return true;

}

return false;

}

int main (void) {

int kase;

scanf("%d", &kase);

while (kase--) {

long long p;

scanf("%lld", &p);

long long ret = p - 1, q = p - 1;

while (check(q)) {

ret = mul(ret, inv(q, p - 2, p), p);

q --;

}

printf("%lld\n", ret);

}

}

1 1000000007 -》328400734

找比p小的最大的素数，阶乘后对p取模