Mục lục nội dung

[Bài thực hành số 03 – Tuần 37 3](#_Toc161238431)

[BÀI TẬP TRÊN LAP 3](#_Toc161238432)

[**Bài tập 3.1.**Dãy Lucas được định nghĩa bởi Ln = Ln-1 + Ln-2 với L0 = 2, L1 = 1. Hãy viết hàm tính số Lucas thứ n. 3](#_Toc161238433)

[**Bài tập 3.2.** Trên bàn cờ vua kích thước n\*n có một quân mã đang ở ô (1, 1). Hãy đưa ra một dãy các di chuyển của mã sao cho mỗi ô trên bàn cờ đều được đi qua đúng 1 lần (ô (1, 1) được xem là đã đi qua). 5](#_Toc161238434)

[**Bài tập 3.3.** Một người xuất phát tại thành phố 1, muốn đi thăm tất cả các thành phố khác, mỗi thành phố đúng 1 lần và quay về 1. Chi phí để đi từ thành phố i sang thành phố j là cij. Hãy tìm tổng chi phí nhỏ nhất có thể **.**. 7](#_Toc161238435)

[**Bài tập 3.4.** Cho dãy a có n phần tử. Một dãy con của  a là dãy thu được bằng cách xóa đi một số phần tử của  a và giữ nguyên thứ tự các phần tử còn lại (có thể không xóa phần tử nào). Hãy tìm dãy con tăng dài nhất của a7](#_Toc161238436)……………………………………9

[**Bài tập 3.5.** Tính hệ số tổ hợp C(n, k)……………………………………………….12](#_Toc161238437)

[**Bài tập 3.6.** Tìm ước chung lớn nhất 7](#_Toc161238438)

[**Bài tập 3.7.** Liệt kê xâu nhị phân 8](#_Toc161238439)

[**Bài tập 3.8.** Đếm cân 8](#_Toc161238440)

[**BÀI TẬP VỀ NHÀ** 8](#_Toc161238441)

# **BÀI THỰC HÀNH SỐ 02 – TUẦN 35**

# BÀI TẬP TRÊN LAP

**Bài tập 3.1.** Dãy Lucas được định nghĩa bởi Ln = Ln-1 + Ln-2 với L0 = 2, L1 = 1. Hãy viết hàm tính số Lucas thứ n.

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động#include<bits/stdc++.h>

using namespace std;

int lucas(int n){

if (n == 0) return 2;

else if (n == 1) return 1;

else return lucas(n - 1) + lucas(n - 2);

}

## **Bài tập 3.2.** Trên bàn cờ vua kích thước n\*n có một quân mã đang ở ô (1, 1). Hãy đưa ra một dãy các di chuyển của mã sao cho mỗi ô trên bàn cờ đều được đi qua đúng 1 lần (ô (1, 1) được xem là đã đi qua).

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động#include<bits/stdc++.h>

using namespace std;

int n;

int X[100], Y[100];

int mark[100][100];

const int hx[] = {1,1,2,2,-1,-1,-2,-2};

const int hy[] = {2,-2,1,-1,2,-2,1,-1};

void print\_sol(){

for (int i = 1; i <= n\*n; i++){

printf("(%d %d)\n", X[i], Y[i]);

}

exit(0);

}

void Try(int k){

for (int i = 0; i < 8 ; i++){

int xx = X[k-1] + hx[i];

int yy = Y[k-1] + hy[i];

if (xx >= 1 && xx <= n && yy >= 1 && yy <= n && mark[xx][yy] == 0){

X[k] = xx;

Y[k] = yy;

mark[xx][yy] = 1;

if (k == n\*n) print\_sol();

else Try(k + 1);

mark[xx][yy] = 0;

}

}

}

int main(){

std::cin>>n;

mark[1][1] = 1;

X[1] = Y[1] = 1;

Try(2);

return 0;

}

## **Bài tập 3.3.** Một người xuất phát tại thành phố 1, muốn đi thăm tất cả các thành phố khác, mỗi thành phố đúng 1 lần và quay về 1. Chi phí để đi từ thành phố i sang thành phố j là cij. Hãy tìm tổng chi phí nhỏ nhất có thể

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

## **Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm Mô tả được tạo tự động**

## #include<bits/stdc++.h>

## using namespace std;

## #define MAX 100

## int n, c[MAX][MAX];

## int cmin = INT\_MAX;

## int best = INT\_MAX;

## int curr;

## int mark[MAX];

## int x[MAX];

## void input(){

## std::cin>>n;

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

## for (int j = 1; j <= n ; j++){

## std::cin>>c[i][j];

## if (c[i][j]) cmin = std::min(cmin, c[i][j]);

## }

## }

## }

## void Try(int k){

## if (curr + cmin\*(n-k+1) >= best) {

## return;

## }

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

## if (mark[i] == 0){

## mark[i] = 1;

## x[k] = i;

## curr += c[x[k-1]][i];

## if (k == n){

## if (curr + c[x[n]][1] < best)

## best = curr + c[x[n]][1];

## }

## else Try(k + 1);

## mark[i] = 0;

## curr -= c[x[k-1]][i];

## }

## }

## }

## int main(){

## /\*

## Nguyen Duy Khanh - 20225019

## Ma lop: 738919

## \*/

## input();

## x[1] = 1;

## Try(2);

## std::cout<<best;

## return 0;

}

## **Bài tập 3.4.** Cho dãy a có  n phần tử. Một dãy con của  a là dãy thu được bằng cách xóa đi một số phần tử của  a và giữ nguyên thứ tự các phần tử còn lại (có thể không xóa phần tử nào). Hãy tìm dãy con tăng dài nhất của a.

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động#include<bits/stdc++.h>

using namespace std;

int a[1000], n;

int mem[1000];

void init(){

## memset(mem, -1, sizeof(mem));

## }

## int lis(int i){

## if (mem[i] != -1) return mem[i];

## mem[i] = 1;

## for (int j = 0; j < i ; j++){

## if (a[j] < a[i])

## mem[i] = max(mem[i], lis(j) + 1);

## }

## return mem[i];

## }

## void trace(int i){

## for (int j = 0 ; j < i ; j++){

## if ((a[j] < a[i]) && mem[i] == 1 + mem[j]){

## trace(j);

## break;

## }

## }

## std::cout<<a[i]<<" ";

## }

## int main(){

## init();

## std::cin>>n;

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

## std::cin>>a[i];

## }

## int res = 1;

## int pos = 0;

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

## if (res < lis(i)){

## res = lis(i);

## pos = i;

## }

## }

## std::cout<<res<<std::endl;

## trace(pos);

## return 0;

## }

## **Bài tập 3.5.** Tính hệ số tổ hợp C(n, k)

Ảnh có chứa văn bản, màn hình, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

#include<bits/stdc++.h>

using namespace std;

int binom(int n, int k){

if (k > n) return 0;

if (k == 0) return 1;

return binom(n-1, k) + binom(n-1, k-1);

}

int binom2(int n, int k){

int c[1001][1001];

c[1][0] = 1;

c[1][1] = 1;

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

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

if (j == 0 || j == i) c[i][j] = 1;

else c[i][j] = c[i-1][j-1] + c[i-1][j];

}

}

return c[n][k];

}

int main(){

/\*

Nguyen Duy Khanh - 20225019

Ma lop: 738919

\*/

int m;

std::cin>>m;

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

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

printf("%d ", binom(n,k));

}

printf("\n");

}

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

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

printf("%d ", binom2(n,k));

}

printf("\n");

}

return 0;

}

**Bài tập 3.6.** Tìm ước chung lớn nhất của hai số nguyên a, b cho trước.  
 Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động#include<iostream>

using namespace std;

int gcd(int a, int b){

if (b == 0) return a;

return gcd(b, a % b);

}

int gcd2(int a, int b){

if (b == 0) return a;

while (a != b){

if (a > b) a -= b;

else {

b -= a;

}

}

return a;

}

int main(){

int a,b;

std::cin>>a>>b;

std::cout<<gcd(a,b)<<std::endl<<gcd2(a,b);

return 0;

}

**Bài tập 3.7.** Sử dụng phương pháp khử đệ quy bằng stack, hãy liệt kê các xâu nhị phân độ dài n không có k bit 1 nào liên tiếp

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, số

Mô tả được tạo tự động#include <bits/stdc++.h>

using namespace std;

struct state{

int i, j, old\_L;

//# constructor

state(int \_i = 0, int \_j = 0, int \_L = 0):

i(\_i), j(\_j), old\_L(\_L){}

};

int main() {

/\*

Nguyen Duy Khanh - 20225019

Ma lop - 738919

\*/

int n, k;

cin >> n >> k;

int x[n+1];

stack<state> s;

int L = 0;

s.push(state(1, 0));

while (!s.empty()){

state &top = s.top();

if (top.i > n){

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

cout << x[i] << " \n"[i == n];

s.pop();

continue;

}

if(top.j > 0)

L = top.old\_L;

if(top.j >1){

s.pop();

continue;

}

if(L + 1 < k || top.j == 0)

{

x[top.i] = top.j;

top.old\_L = L;

L = top.j ? L + 1 : 0;

s.push(state(top.i + 1, 0));

}

++top.j;

}

return 0;

}

**Bài tập 3.8.** Cân đĩa:

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

#include <bits/stdc++.h>

using namespace std;

struct state{

int i, j, s;

state(int \_i = 0, int \_j = 0, int \_s=0): i(\_i), j(\_j), s(\_s){}

};

int main() {

int n, M;

cin >> n >> M;

int m[n+1];

for (int i = 1; i <= n; ++i) cin >> m[i];

int x[n+1];

stack<state> s;

//# sum of selected weights

int sum = 0;

s.push(state(1, -1, -m[1]));

s.push(state(1, 0, 0));

s.push(state(1, 1, m[1]));

while (!s.empty()){

state top = s.top();

if (top.i > n){

if (top.s == M){

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

if (x[i] == -1) cout << '-' << m[i];

if (x[i] == 1) cout << '+' << m[i];

}

cout << "=" << M;

exit(0);

}

s.pop();

continue;

}

s.pop();

x[top.i]=top.j;

s.push(state(top.i+1, -1, top.s-m[top.i+1]));

s.push(state(top.i+1, 0,top.s));

s.push(state(top.i+1, 1,top.s+m[top.i+1]));

}

cout << -1;

return 0;

}

## **Bài tập về nhà:**

Bài 3.9. Một y tá cần lập lịch làm việc trong *N*ngày, mỗi ngày chỉ có thể là làm việc hay nghỉ ngơi. Một lịch làm việc là tốt nếu không có hai ngày nghỉ nào liên tiếp và mọi chuỗi ngày tối đại làm việc liên tiếp đều có số ngày thuộc đoạn [*K*1,*K*2]. Hãy liệt kê tất cả các cách lập lịch tốt, với mỗi lịch in ra trên một dòng một xâu nhị phân độ dài *n* với bit 0/1 tương ứng là nghỉ/làm việc. Các xâu phải được in ra theo thứ tự từ điển

Ảnh có chứa ảnh chụp màn hình, văn bản, màu đen

Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

#include<bits/stdc++.h>

using namespace std;

const int MAX = 1000;

const int du = 1000000000 + 7;

int n, k1, k2;

int x[MAX];

int cnt = 0;

int so0 = 0, so1 = 0;

void inputData(){

cin >> n >> k1 >> k2;

}

bool check(int a, int i){

if(a==1) return true;

else {

if(i==0){

if(x[a-1] == 0) return false;

else {

if(so1<k1) return false;

}

} else {

if(x[a-1] == 0){

if(n-a+1 < k1) return false;

} else {

if(so1>=k2) return false;

}

}

return true;

}

}

void solution(){

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

cout << x[i];

cout << endl;

cnt++;

}

void TRY(int a){

for(int i=0; i<=1; i++){

if(check(a,i)){

x[a] = i;

int pre = so1;

if(i == 1){

if(x[a-1] == 1) so1++;

else so1 = 1;

} else {

so1 = 0;

}

if(a==n) solution();

else TRY(a+1);

so1 = pre;

}

}

}

int main(){

/\*

Nguyen Duy Khanh - 20225019

Ma lop: 738919

\*/

inputData();

TRY(1);

return 0;

}

## **Bài tập 3.10:** Khoảng cách Hamming:

#include<bits/stdc++.h>

using namespace std;

const int MAX = 20;

int N, H;

int x[MAX];

int S[MAX];

int cnt = 0;

void input(){

    cin >> N >> H;

    for(int i=0; i<N; i++)

        S[i] = 0;

    for(int i=0; i<N; i++)

        x[i] = 0;

}

int checkHamming(int str1[], int str2[]){

    int lens = N;

    int cnt = 0;

    for(int i=0; i<lens; i++){

        if(str1[i] != str2[i]) cnt++;

    }

    return cnt;

}

bool check(int a, int i){

    return true;

}

void solution(){

    if(checkHamming(x,S) == H){

        for(int i=0; i<N; i++)

            cout << x[i];

        cout << endl;

    }

}

void TRY(int a){

    for(int i=0; i<=1; i++){

        x[a] = i;

        if(a == N-1) solution();

        else

            TRY(a+1);

    }

}

int main(){

    /\*

    Nguyen Duy Khanh - 20225019

    Ma lop: 738919

    \*/

    int T;

    cin >> T;

    while(T > 0){

        input();

        TRY(0);

        T--;

    }

}

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình

Mô tả được tạo tự động

#include<bits/stdc++.h>

using namespace std;

const int MAX = 20;

int N, H;

int x[MAX];

int S[MAX];

int cnt = 0;

void input(){

cin >> N >> H;

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

S[i] = 0;

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

x[i] = 0;

}

int checkHamming(int str1[], int str2[]){

int lens = N;

int cnt = 0;

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

if(str1[i] != str2[i]) cnt++;

}

return cnt;

}

bool check(int a, int i){

return true;

}

void solution(){

if(checkHamming(x,S) == H){

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

cout << x[i];

cout << endl;

}

}

void TRY(int a){

for(int i=0; i<=1; i++){

x[a] = i;

if(a == N-1) solution();

else

TRY(a+1);

}

}

int main(){

/\*

Nguyen Duy Khanh - 20225019

Ma lop: 738919

\*/

int T;

cin >> T;

while(T > 0){

input();

TRY(0);

T--;

}

}

**Bài tập 3.11**: Lịch trình chụp ảnh

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, màn hình

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, số

Mô tả được tạo tự động

#include<bits/stdc++.h>

using namespace std;

const int MAX = 10000;

int n, r;

int price[MAX][MAX];

int x[MAX];

bool visited[MAX];

vector<int> vt;

int best, sum\_price, start, des, numOfPoint;

void inputData(){

cin >> n >> r;

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

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

cin >> price[i][j];

}

}

}

bool check(int a, int i){

if(visited[vt[i]]) return false;

if(price[x[a-1]][vt[i]] == 0) return false;

return true;

}

void solution(){

if (price[x[numOfPoint-2]][des] == 0) return;

best = min(best, sum\_price + price[x[numOfPoint-2]][des]);

}

void TRY(int k){

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

if (check(k, i)){

visited[vt[i]] = true;

sum\_price += price[x[k-1]][vt[i]];

x[k] = vt[i];

if (k == numOfPoint - 2) solution();

else TRY(k+1);

visited[vt[i]] = false;

sum\_price -= price[x[k-1]][vt[i]];

}

}

}

int main(){

/\*

Nguyen Duy Khanh - 20225019

Ma lop: 738919

\*/

string str;

inputData();

getline(cin,str);

while(r > 0){

best = INT\_MAX;

sum\_price = 0;

getline(cin, str);

while (!str.empty()){

stringstream convert(str.substr(0, str.find(" ")));

int tmp = 0;

convert >> tmp;

vt.push\_back(tmp - 1);

if (str.find(" ") > str.size()) break;

else str.erase(0, str.find(" ") + 1);

}

start = vt[0];

des = vt[vt.size() - 1];

numOfPoint = vt.size();

x[0] = start; x[numOfPoint-1] = des;

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

visited[i] = false;

}

TRY(1);

if (best == INT\_MAX) cout << "0" << endl;

else cout << best << endl;

vt.erase(vt.begin(), vt.end());

r--;

}

}

## **Bài tập 3.12:** Đếm đường đi

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, phần mềm, số

Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động#include<bits/stdc++.h>

using namespace std;

const int MAX = 100;

int n, k;

int m;

vector<vector<int> > vt;

int x[MAX];

bool visited[MAX];

int res;

void input(){

cin >> n >> k;

cin >> m;

vt.resize(n+1);

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

int tmp1, tmp2;

cin >> tmp1 >> tmp2;

vt[tmp1-1].push\_back(tmp2-1);

vt[tmp2-1].push\_back(tmp1-1);

}

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

visited[i] = false;

}

res = 0;

}

bool check(int a, int i){

if(a == 0) return true;

if(visited[i]) return false;

int index = 0;

for(int j=0; j<vt[x[a-1]].size(); j++){

if(i == vt[x[a-1]][j]) index++;

}

if(index == 0) return false;

return true;

}

void solution(){

res++;

}

void TRY(int a){

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

if(check(a, i)){

visited[i] = true;

x[a] = i;

if(a == k) solution();

else TRY(a+1);

visited[i] = false;

}

}

}

int main(){

/\*

Nguyen Duy Khanh - 20225019

Ma lop: 738919

\*/

input();

TRY(0);

cout << res / 2;

}