Nguyễn Xuân Huy

Viện CNTT Viện HL KH & CN VN

0903203800 nxhuy564@gmail.com

**ADVANCED ALGORITHMS AND APPLICATIONS**

**CONTEST 3**

**PTIT 2023**

Contents

[Contest 3 3](#_Toc139548388)

[NHÌN VÀ ĐỌC 3](#_Toc139548389)

[LIỆT KÊ DÃY CON 3](#_Toc139548390)

[CHỌN SỐ TỪ MA TRẬN VUÔNG CẤP N 3](#_Toc139548391)

[ĐẾM TỪ 3](#_Toc139548392)

[ĐẾM DÃY 4](#_Toc139548393)

[GẤP ĐÔI DÃY SỐ 4](#_Toc139548394)

[XẾP HẬU 4](#_Toc139548395)

[XÂU CON CHUNG DÀI NHẤT 4](#_Toc139548396)

[DÃY CON TĂNG DÀI NHẤT 5](#_Toc139548397)

[DÃY CON CÓ TỔNG S 5](#_Toc139548398)

[SỐ NHỎ NHẤT CÓ N ƯỚC SỐ 5](#_Toc139548399)

[ĐƯỜNG ĐI VỚI CHI PHÍ NHỎ NHẤT 5](#_Toc139548400)

[Bài giải Contest 3 5](#_Toc139548401)

[NHÌN VÀ ĐỌC 5](#_Toc139548402)

[LIỆT KÊ DÃY CON 6](#_Toc139548403)

[CHỌN SỐ TỪ MA TRẬN VUÔNG CẤP N 8](#_Toc139548404)

[ĐẾM TỪ 10](#_Toc139548405)

[ĐẾM DÃY 11](#_Toc139548406)

[GẤP ĐÔI DÃY SỐ 12](#_Toc139548407)

[XẾP HẬU 13](#_Toc139548408)

[XÂU CON CHUNG DÀI NHẤT 15](#_Toc139548409)

[DÃY CON TĂNG DÀI NHẤT 18](#_Toc139548410)

[DÃY CON CÓ TỔNG S 20](#_Toc139548411)

[SỐ NHỎ NHẤT CÓ N ƯỚC SỐ 21](#_Toc139548412)

[Hàm Tau 21](#_Toc139548413)

[All Tau 22](#_Toc139548414)

[Tau200K 23](#_Toc139548415)

[TauN 25](#_Toc139548416)

[ĐƯỜNG ĐI VỚI CHI PHÍ NHỎ NHẤT 28](#_Toc139548417)

## Contest 3

### NHÌN VÀ ĐỌC

Phép biến đổi "nhìn và đọc" được định nghĩa như sau: Bắt đầu từ số đầu tiên X.

Với các số giống nhau thì nhóm lại thành một dãy (giả sử có Y số) thì dãy này được chuyển thành YX.

Ví dụ 122344111 đọc là "một 1, hai 2, một 3, hai 4, ba 1", do đó chuỗi 122344111 đượ

chuyển thành 1122132431.

Dữ liệu vào: Dòng đầu chứa số bộ test:

Mỗi dòng sau chứa 1 bộ test chứa chuỗi cần biến đổi (không quá 1000 chữ số)

Kết quả: Với mỗi bộ test, in ra kết quả phép biến đổi

Ví dụ

|  |  |  |
| --- | --- | --- |
| LR.INP | LR.OUT | *Giải thích* |
| 3  122344111  1111111111  12345 | 1122132431  101  1112131415 | *3 test* |

### LIỆT KÊ DÃY CON

Cho dãy A gồm N số tự nhiên khác nhau và số tự nhiên K (2 ≤ K ≤ N ≤ 15, 0 < A[i ]<1000).

Hãy liệt kê mọi dãy con có tổng bằng k của dãy a. Một dãy con của dãy A là dãy nhận được

từ A sau khi xóa bớt một số phần tử không nhất thiết đứng liên tiếp trong A.

Gi[í hạn: 2 < N < 15; 0 < A[i] < 1000.

|  |  |  |
| --- | --- | --- |
| SEQ.INP | SEQ.OUT | *Giải thích* |
| 7 50  5 10 15 20 25 30 35 | 7  20 30  15 35  10 15 25  5 20 25  5 15 30  5 10 35 5  10 15 20 | *7 dãy con* |

### CHỌN SỐ TỪ MA TRẬN VUÔNG CẤP N

Cho ma trận vuông C cấp N gồm N^2 số tự nhiên và số tự nhiên K. Hãy viết chương trình lấy mỗi hàng, mỗi cột duy nhất một phần tử sao cho tổng các phần tử này đúng bằng K.

Dữ liệu vào: Dòng đầu tiên ghi hai số N và K.

N dòng tiếp theo ghi các dòng của ma trận C.

Kết quả: dòng đầu ghi số cách tìm được.

Mỗi dòng tiếp theo ghi một cách theo vị trí của số đó trong lần lượt từng hàng của ma trận.

Giới hạn: 1 ≤ N ≤ 15, các số tự nhiên trong ma trận C có thể trùng nhau và không vượt quá 1000.

0 ≤ K < 10^5.

|  |  |  |
| --- | --- | --- |
| MATRIX.INP | MATRIX.OUT | *Giải thích* |
| 3 10  2 4 **3**  1 **3** 6  **4** 2 4 | 2  1 3 2  3 2 1 | *2 cách*  C[1][1] + C[2][3] + C[3][2] = 2 + 6 + 2 = 10  C[1][3] + C[2][2] + C[3][1] = 3 + 3 + 4 = 10 |

### ĐẾM TỪ

Cho luồng vào gồm không quá 10^6 từ, mỗi từ chỉ bao gồm gồm các chữ cái viết hoa không dấu và có độ dài

không quá 20 ký tự. Các từ phân cách bởi các khoảng trống hoặc dấu xuống dòng.

Hãy đếm xem có bao nhiêu từ khác nhau, mỗi từ xuất hiện bao nhiêu lần.

Kết quả được liệt kê theo thứ tự từ điển.

Ví dụ

|  |  |  |
| --- | --- | --- |
| WORDS.INP | WORDS.OUT | *Giải thích* |
| WHILE DO END BEGIN DO INT  END BEGIN CHAR DO DO REPEAT  INT DO INT BEGIN DO REPEAT  WHILE DO END BEGIN END BEGIN | 7  BEGIN 5  CHAR 1  DO 7  END 4  INT 3  REPEAT 2  WHILE 2 | *có 7 từ khác nhau*  BEGIN *xuất hiện 5 lần*  *…* |

### ĐẾM DÃY

Cho số nguyên dương n. Hãy cho biết có bao nhiêu dãy số nguyên dương

có tổng các phần tử trong dãy bằng n.

Dữ liệu vào: dòng đầu tiên chứa số nguyên T là số bộ dữ liệu test (không quá 20),

mỗi bộ dữ liệu ghi một số nguyên dương n duy nhất không qua 10^18 .

Kết quả: Mỗi bộ dữ liệu ghi ra một số nguyên duy nhất là số dư của kết quả tìm được

khi chia cho 123456789.

Ví dụ

|  |  |  |
| --- | --- | --- |
| SEQ.INP | SEQ.OUT | *Giải thích* |
| 1  3 | 4 | *1 test*  *Có 4 dãy số nguyên dương có tổng 3 là:*  (3), (1,2), (2,1), (1,1,1) |

### GẤP ĐÔI DÃY SỐ

Một dãy số tự nhiên bắt đầu bởi chữ số 1 và được thực hiện N-1 phép biến đổi “gấp đôi” dãy số như sau:

Với dãy số A hiện tại, dãy số mới có dạng A|x|A trong đó x là số tự nhiên bé nhất chưa xuất hiện trong A.

Ví dụ với 2 bước biến đổi, ta có [1] → [1 2 1] → [1 2 1 3 1 2 1]. Hãy xác định số thứ K trong dãy số

cuối cùng là bao nhiêu?

Input: Dòng đầu tiên là số lượng bộ test T (T <= 20).

Mỗi test gồm số nguyên dương N và K (1 ≤ N ≤ 50, 1 ≤ K ≤ 2^N - 1).

Output: Với mỗi test, in ra đáp án trên một dòng.

|  |  |  |
| --- | --- | --- |
| SEQ.INP | SEQ.OUT | *Giải thích* |
| 2  3 2  4 8 | 2  4 | 2 test  Test 1: [1] → [1 2 1] → [1 **2** 1 3 1 2 1].  Test 2: [1] → [1 2 1] → [1 2 1 3 1 2 1]  → [1 2 1 3 1 2 1 **4** 1 2 1 3 1 2 1 ] |

### XẾP HẬU

Cho một bàn cờ 8 x 8, mỗi ô có một giá trị A[i][j] nhất định (0 <= A[i][j] <= 100), tương ứng

với điểm số đạt được nếu như bạn đặt một quân cờ vào đó.

Nhiệm vụ của bạn là đặt 8 quân hậu lên bàn cờ, sao cho không có 2 quân nào tự ăn nhau,

và số điểm đạt được là lớn nhất.

Input: Dòng đầu tiên là số lượng bộ test T (T <= 20).

Mỗi test gồm 8 dòng, mỗi dòng 8 số nguyên mô tả bàn cờ.

Output: Với mỗi test, in ra đáp án trên một dòng.

Ví dụ: Input

1 2 3 4 5 6 7 8

9 10 11 12 13 14 15 16

17 18 19 20 21 22 23 24

25 26 27 28 29 30 31 32

33 34 35 36 37 38 39 40

41 42 43 44 45 46 47 48

49 50 51 52 53 54 55 56

57 58 59 60 61 62 63 64

Output: 260

### XÂU CON CHUNG DÀI NHẤT

Xâu ký tự X được gọi là xâu con của xâu ký tự Y nếu ta có thể xoá đi một số ký tự trong xâu Y để

thu được xâu X.

Cho hai xâu ký tự A và B, hãy cho biết độ dài của xâu con chung dài nhất của A và B.

Dữ liệu vào gồm 2 dòng: Dòng 1: chứa xâu A. Dòng 2: chứa xâu B. Mỗi xâu dài không quá 2000 ký tự.

Kết quả: Chỉ gồm một dòng ghi độ dài xâu C tìm được

|  |  |  |
| --- | --- | --- |
| COMSEQ.INP | OUT | *Giải thích* |
| abc1def2ghi3  abcdefghi123 | 10 | xâu con chung là  abcdefghi3 |

### DÃY CON TĂNG DÀI NHẤT

Cho một dãy số nguyên gồm n phần tử . Biết rằng dãy con tăng đơn điệu

của dãy *a là dãy*  thỏa điều kiện và .

Hãy cho biết dãy con tăng đơn điệu dài nhất của dãy này có bao nhiêu phần tử? 1 ≤ n ≤ 1000.

### DÃY CON CÓ TỔNG S

Cho dãy a gồm n số nguyên dương. Có hay không một dãy con của dãy a (không nhất thiết là các

phần tử liên tiếp trong dãy) có tổng bằng s cho trước ?

Giới hạn: 1 ≤ n ≤ 200; 1 ≤ s ≤ 4000.

### SỐ NHỎ NHẤT CÓ N ƯỚC SỐ

Cho số nguyên dương N. Nhiệm vụ của bạn là tìm số K nhỏ nhất, sao cho K có đúng N ước số. Input đảm bảo rằng đáp án không vượt quá 10^18. Dữ liệu vào: Dòng đầu tiên là số lượng bộ test T (T <= 10). Mỗi test gồm 1 số nguyên N ( 1 <= N <= 1000). 5 Kết quả: Với mỗi test, in ra đáp án trên một dòng.

### ĐƯỜNG ĐI VỚI CHI PHÍ NHỎ NHẤT

Cho ma trận A kích thước N x M (N hàng, M cột). Bạn được phép đi sang trái, sang phải và

đi xuống ô chéo dưới. Khi đi qua ô (i, j), điểm nhận được sẽ là A[i][j].

Hãy tìm đường đi từ ô (1, 1) tới ô (N, M) sao cho tổng điểm là nhỏ nhất.

Input: Dòng đầu tiên là số lượng bộ test T (T <= 20).

Mỗi test gồm số nguyên dương N và M (không quá 1000).

N dòng tiếp theo, mỗi dòng gồm M số nguyên A[i][j] (0 <= A[i] <= 1000).

Output: Với mỗi test, in ra độ dài dãy con tăng dài nhất trên một dòng.

Ví dụ:

|  |  |  |  |
| --- | --- | --- | --- |
|  | *1* | *2* | *3* |
| *1* | **1** | **2** | 3 |
| *2* | 4 | 8 | **2** |
| *3* | 1 | 5 | **3** |

Input

1

3 3

1 2 3

4 8 2

1 5 3

Output: 8

Giải thích test: Đường đi (1, 1) → (1, 2) → (2, 3) → (3, 3).

## Bài giải Contest 3

### NHÌN VÀ ĐỌC

**Algorithm**

<số lượng> <Giá trị>

**Complexity**: O(n)

**Program**

/\*

Name: LR.CPP

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description: Nhin va doc (Looking and Reading)

\*/

#include <bits/stdc++.h>

using namespace std;

const char \* fn = "LR.INP";

const char \* gn = "LR.OUT";

void Go() {

cout << " ? ";

fflush(stdin);

if(cin.get() == '.') exit(0);

}

string Str(int x) {

stringstream ss;

ss << x;

return ss.str();

}

/\*-----------------------------

<num> <val>

122344111 -> 11 22 13 24 31

1111111111

12345

-------------------------------\*/

string Reading(string s) {

string w = "";

int j = 0, num = 1;

for(int i = 1; i < s.length(); ++i) {

if (s[i] == s[j]) {

++num;

}

else {

w += Str(num) + s[j];

num = 1;

j = i;

}

}

w += Str(num) + s[j];

return w;

}

void Run() {

ifstream f(fn);

if(f.fail()) {

cerr << "\n No input file " << fn;

exit(0);

}

int q; // so test

f >> q;

cout << "\n " << q << " test";

string s, w;

ofstream g(gn);

if(g.fail()) {

cerr << "\n Can't create output file " << gn;

exit(0);

}

for(int t = 1; t <= q; ++t) {

f >> s;

w = Reading(s);

cout << "\n Test No " << t << ": " << s << " -> " << w;

g << w << endl;

}

f.close(); g.close();

}

main() {

Run();

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### LIỆT KÊ DÃY CON

**Algorithm**

Vét cạn mọi chỉnh hợp của n

**Complexity**: O(2^n)

**Program**

/\*

Name: SUBSEQ.CPP

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description: Liet ke cac day con co tong k

input: SEQ.INP

7 50

5 10 15 20 25 30 35

output: SEQ.OUT

7

5 10 15 20

10 15 25

5 20 25

5 15 30

20 30

5 10 35

15 35

\*/

#include <bits/stdc++.h>

using namespace std;

const char \* fn = "SEQ.INP";

const char \* gn = "SEQ.OUT";

const int MN = 15;

int \*a;

bitset<MN> b;

vector<int> v;

void Go() {

cout << " ? ";

fflush(stdin);

if(cin.get() == '.') exit(0);

}

// hien thi a theo b

void Print(int n, const char \* msg = "") {

cout << msg;

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

if (b[i]) cout << " " << a[i];

}

// Tinh tong theo b

int Sum(int n) {

int t = 0;

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

if (b[i]) t += a[i];

return t;

}

void Run() {

ifstream f(fn);

if(f.fail()) {

cerr << "\n No input file " << fn;

exit(0);

}

int n, k;

f >> n >> k;

cout << "\n n = " << n << " k = " << k;

a = new int[n];

for(int i = 0; i < n; ++i) f >> a[i];

cout << "\n a: ";

for(int i = 0; i < n; ++i) cout << " " << a[i];

f.close(); //g.close();

int maxn = 1 << n;

cout << "\n max n = " << maxn;

int d = 0;

v.clear();

for(int x = 0; x < maxn; ++x) {

b = x;

if (Sum(n) == k) {

++d;

v.push\_back(x); // save to v

}

}

cout << "\n Total: " << d;

ofstream g(gn);

if(f.fail()) {

cerr << "\n Can't create output file " << gn;

exit(0);

}

g << d << endl;

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

b = v[i];

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

if (b[j]) g << a[j] << " ";

g << endl;

}

g.close();

}

main() {

Run();

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### CHỌN SỐ TỪ MA TRẬN VUÔNG CẤP N

**Algorithm**

Vét cạn: Duyệt các hoán vị h:1,,n. h(i) cho biết tại dòng I cần lấy phần tử h(i) .

**Complexity**: **O(n!)**

**Program**

/\*

Name: MATRIX.CPP

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description: Chon duy nhat 1 dong, 1 cots cho tong k

input: MATRIX.INP

3 10

2 4 3

1 3 6

4 2 4

output: MATRIX.OUT

2

1 3 2

3 2 1

\*/

#include <bits/stdc++.h>

using namespace std;

const char \* fn = "MATRIX.INP";

const char \* gn = "MATRIX.OUT";

const int MN = 15;

int \*\*a;

vector<string> v;

int \*h;

void Go() {

cout << " ? ";

fflush(stdin);

if(cin.get() == '.') exit(0);

}

// Tinh tong theo b

int Sum(int n) {

int t = 0;

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

t += a[i][h[i]];

}

// cout << "\n Sum = " << t;

return t;

}

// Hien thi hoan vi h

int Print(int n, const char \* msg = "") {

cout << msg;

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

cout << " " << h[i];

}

}

// Tim hoan vi tiep theo cua h

bool Next(int n) {

int i, j;

for(i = n-1; i > 0; --i)

if(h[i] < h[i+1]) break;

if (i < 1) return false;

for(j = n; j > i; --j)

if(h[j] > h[i]) break;

int x = h[i]; h[i] = h[j]; h[j] = x;

++i; j = n;

while(i < j) {

x = h[i]; h[i] = h[j]; h[j] = x;

++i; --j;

}

return true;

}

// chuyen h sang string

string Str(int n) {

stringstream ss;

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

ss << h[i] << " ";

}

return ss.str();

}

void Run() {

ifstream f(fn);

if(f.fail()) {

cerr << "\n No input file " << fn;

exit(0);

}

int n, k;

f >> n >> k;

cout << "\n Input n = " << n << " k = " << k;

a = new int \* [n+1];

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

a[i] = new int[n+1];

}

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

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

f >> a[i][j];

f.close();

cout << "\n a: " ;

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

cout << endl;

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

cout << " " << a[i][j];

}

// Khoi tao hoan vi h

h = new int[n+1];

for(int i = 0; i <= n; ++i) h[i] = i;

int d = 0; // dem cac phuong an

v.clear();

do {

if (Sum(n) == k) {

++d;

v.push\_back(Str(n));

Print(n, "\n h: "); // Go();

}

} while(Next(n));

cout << "\n Total: " << d;

// ghi file

ofstream g(gn);

if(f.fail()) {

cerr << "\n Can't create output file " << gn;

exit(0);

}

g << d << endl;

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

g << v[i] << endl;

}

g.close();

}

main() {

Run();

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### ĐẾM TỪ

**Algorithm**

* Đọc từng dòng
* Tách từ
* sort
* Đém các từ giống nhau (đứng cạnh nhau)
* Ghi file

**Complexity**: O(n \* log n) chi phí sort, n là số lượng từ

**Program**

/\*

Name: WORDS.CPP

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description: Dem tu

input: WORDS.INP

WHILE DO END BEGIN DO INT

END BEGIN CHAR DO DO REPEAT

INT DO INT BEGIN DO REPEAT

WHILE DO END BEGIN END BEGIN

output: WORDS.OUT

7

BEGIN 5

CHAR 1

DO 7

END 4

INT 3

REPEAT 2

WHILE 2

\*/

#include <bits/stdc++.h>

using namespace std;

const char \* fn = "WORDS.INP";

const char \* gn = "WORDS.OUT";

void Go() {

cout << " ? ";

fflush(stdin);

if(cin.get() == '.') exit(0);

}

bool ABC(char c) {

return (c >= 'A' && c <= 'Z');

}

string Str(int d) {

stringstream ss;

ss << d;

return ss.str();

}

void Run() {

ifstream f(fn);

if(f.fail()) {

cerr << "\n No input file " << fn;

exit(0);

}

string s, w;

vector<string> v;

while(getline(f,s)) {

// doc 1 dong

if (s == "") break;

cout << "\n Line: |" << s << "|";

stringstream ss(s);

// tach tu

while(getline(ss,w,' ')) {

if (w != "") v.push\_back(w);

}

}

f.close();

cout << "\n " << v.size();

sort(v.begin(), v.end());

cout << "\n Sorted v: ";

for(int i = 0; i < v.size(); ++i)

cout << "\n" << v[i];

v.push\_back("."); // them linh canh vao cuoi v

vector<string> vv;

int j = 0, d = 1;

for(int i = 1; i < v.size(); ++i) {

if (v[i] == v[j]) ++d;

else {

vv.push\_back(v[j] + " " + Str(d));

d = 1;

j = i;

}

}

ofstream g(gn); // ghi file

if(g.fail()) {

cerr << "\n Can't create output file " << gn;

exit(0);

}

g << vv.size() << endl;

for(int i = 0; i < vv.size(); ++i) {

g << vv[i] << endl;

}

g.close();

}

main() {

Run();

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### ĐẾM DÃY

**Algorithm**

Gọi C(n) là số dãy con gồm các số nguyên dương có tổng n.

C(1) = 1; có 1 dãy duy nhất là (1)

C(2) = 2; có 2 dãy: (2) , (1,1)

C(3) = 4; có 4 dãy: (3), (2,1), (1,2), (1,1,1)

…

Ta chứng minh .

Ta có, khi n = 1: .

Giả sử với mọi i : 0.. n ta có . Ta cần chứng minh .

Ta phân loại các các dãy của C(n+1) thành các lớp không giao nhau với phần tử d đầu dãy như sau:

d = 1: còn lại tổng n tạo thành C(n) dãy.

d = 2: còn lại tổng n-1 tạo thành C(n) dãy.

…

d = n: còn lại tổng n+1-n = 1 tạo thành C(1) dãy.

Tổng công ta có: C(n+1) = C(1) + C(2) +…+ C(n) = , điều cần chứng minh.

Vậy đpá số là

**Complexity**: O(1)

Một cách tiếp cận khác như sau:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *index* | *0* | *1* | *2* | *3* | *4* | *5* | *6* |
| hoa | \* | \* | \* | \* | \* | \* | \* |
| nhãn | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| số lượng | 2 | | 3 | | | 2 | |

Giả sử có 6 bông hoa, có bao nhiêu cách bó thành dãy các bó hoa liên tiếp nhau?

Một cách làm đơn giản là bạn đặt dãy liên tiếp các số 0/1 vào mỗi bông hoa với ý nghĩa là chọn

liên tiếp các bông hoa có nhãn 1 để bó thành một bó, sau đó bạn chọ liên tiếp các bông hóa có nhãn 0

để tạo bó hoa thứ hai,… Theo các này bạn bó được dãy (2, 3, 2). Với mọi cáh gán nhãn 0/1 như trên

bạn tạo ra dãy, trong số đó có các dãy trùng nhau, ví dụ nhãn 1100011 và 0011100 sẽ tạo ra hai

dãy trùng nhau là (2, 3, 2). Hai nhãn này là số lật của nhau.

Vậy bạn sẽ tạo ra dãy bó hoa khác nhau.

Với n dài 18 chữ số thập phân, ta dùng phương pháp chia để trị để tính hàm C(n) = .

Tổng quát, để tính dựa vào hệ thức sau:

**Program**

/\*

Name: Name: SEQCOUNT.CPP

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description:

Dem cac day so co tong n

2^(n-1) mod 123456789.

\*/

#include <bits/stdc++.h>

using namespace std;

typedef unsigned long long UL;

const UL MN = 123456789;

// 2%n mod MN

int C(UL n) {

UL z = 1; // 2&0 mod MN

UL a = 2;

while(n) {

if((n & 1) == 1) // odd n

z = (z \* a) % MN;

n >>= 1; .. n = n / 2

a = (a\*a) % MN;

}

return z;

}

main() {

for(int n = 1; n <= 500; ++n)

cout << "\n n = " << n << ": " << C(n-1);

UL n = 123456789012345678;

cout << "\n n = " << n << ": " << C(n-1);

//cout << "\n n = " << n << ": " << C(n-1);

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### GẤP ĐÔI DÃY SỐ

**Algorithm**

Tìm quy luật tính len và số giữa của dãy. Gọi C(n) là dãy số được sinh ra sau n bước. Ta có

C(1) = [1], Len(1) = 1, M(1) = 1

C(2) = [1 2 1], Len(2) = 3, M(2) = 2

C(3) = [1 2 1 3 1 2 1], Len(3) = 7, M(3) = 3

…

Tổng quát, với C(n) ta có

Len(n) = 2Len(n-1) + 1, M(n) = n

Day thu n: ---------Len(n-1)---------------N--------- Len(n-1)--------------

Giả sử ta cần tìm số thứ k của dãy;

Nếu k = Len(n-1)+1 thì return n // số ở vị trí giữa

Nếu k > Len(n-1)+1 thì xét số k-(Len(n-1)+1) trong C(n-1)

Nếu k ≤ Len(n-1) thì xét số k trong C(n-1)

**Complexity:** O(n)

**Program**

/\*

Name: Name: SEQ2.CPP (gap doi day so)

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description:

\*/

#include <bits/stdc++.h>

using namespace std;

const int MN = 50; // max N

typedef unsigned long long UL;

UL len[MN+1];

// Tim so thu k trong C(n)

int Find(int n, UL k) {

//(n-1)n(n-1)

// len[n-1]nlen[n-1]

UL m;

// cout << "\n m = " << m;

while(n > 1) {

m = len[n-1] + 1;

// cout << "\n n = " << n << " k = " << k;

if (k == m) return n;

else if (k > m) k -= m;

--n;

}

return n;

}

void SetLen() {

len[1] = 1;

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

len[i] = 2\*len[i-1]+1;

// cout << "\n " << len[i];

}

}

main() {

SetLen();

int n = 4;

for(UL k = 1; k <= len[n]; ++k)

cout << " " << Find(n, k);

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### XẾP HẬU

**Algorithm**

Dữ liệu nhỏ, n = 8 nên vét cạn: xét mọi hoán vị của 1..8

**Complexity**: O(n!), n = số hậu

**Program**

/\*

Name: QUEENS.CPP

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description:

Xep Hau toi uu

\*/

#include <bits/stdc++.h>

using namespace std;

const int MN = 8;

int a[MN+1][MN+1]; // award

int p[MN+1];

int score = 0;

void Go() {

cout << " ? ";

fflush(stdin);

if(cin.get() == '.') exit(0);

}

void ReadData(const char\* fn) {

ifstream f(fn);

if (f.fail()) {

cerr << "\n No flie " << fn;

exit(0);

}

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

cout << endl;

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

f >> a[i][j];

cout << " " << a[i][j];

}

}

f.close();

}

bool Next() {

int i, j;

for(i = MN-1; i > 0; --i)

if(p[i] < p[i+1]) break;

if(i < 1) return false;

for(j = MN; j > i; --j)

if(p[j] > p[i]) break;

int x = p[i]; p[i] = p[j]; p[j] = x;

++i; j = MN;

while(i < j) {

x = p[i]; p[i] = p[j]; p[j] = x;

++i; --j;

}

return true;

}

int GetScore() {

int sc = 0; // score

for(int i = 1; i <= MN; ++i) // Hau i

// Hau i dung tai dong p[i]

// se co thuong a[i][p[i]]

sc += a[i][p[i]];

return sc;

}

bool Sat() {

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

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

if (abs(p[i]-p[j]) == i-j) return false;

}

}

return true;

}

void Run() {

// init permutation p

for(int i = 0; i <= MN; ++i) p[i] = i;

int d = 0, sc;

score = GetScore();

while (Next()) {

if (Sat()) score = max(score, GetScore());

}

cout << "\n Result: " << score;

}

main() {

ReadData("QUEENS.INP");

Run();

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### XÂU CON CHUNG DÀI NHẤT

#### Phương án 1. Đệ quy

**Algorithm**

Len(i,j) = max len của xâu con chung khi xét 2 xâu a(0..i) và b(0..j).

a(i) == b(j): len(I,j) = len(i-1,j-1) + 1

a: ------------x = a(i)

b: ----------------x = b(j)

a(i) != b(j): len(I,j) = max(len(i-1,j), len(I, j-1))

a: ------------x = a(i)

b: ----------------y = b(j)

**Complexity**: Very big

**Program**

/\*

Name: LCS.CPP (Ver. 1: recursive)

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description:

Longest sub string (Xau con chung dai nhat)

A: "Posts and Telecommunications Institute of Technology"

B: "Posts\*and\*Telecommunications\*Institute\*of\*Technology, Hanoi, VietNam"

C: "Posts and Telecommunications Institute of Technology"

Dynamic Programming

Len(i,j) = max len A(0..i), B(0..j)

A(i) = B(j) : Len(i,j) = Len(i-1,j-1) + 1

A(i) != B(j): Len(i,j) = max(Len(i-1,j), len(i, j-1))

------x

----------x

------x

----------y

\*/

#include <bits/stdc++.h>

using namespace std;

string a, b;

void Go() {

cout << " ? ";

fflush(stdin);

if(cin.get() == '.') exit(0);

}

// Recursive

// a(0..i), b(0..j)

int Len(int i, int j) {

if(i < 0 || j < 0) return 0;

return (a[i] == b[j]) ? Len(i-1,j-1) + 1

: max(Len(i, j-1), Len(i-1,j));

}

// Recursive

int Run(string inpa, string inpb) {

a = inpa; b = inpb;

cout << "\n Input a: " << a;

cout << "\n Input b: " << b;

return Len(a.length()-1, b.length()-1);

}

main() {

int lenc = Run("abcd", "a1b2c345d123");

cout << "\n Result: " << lenc;

lenc = Run("Posts and Telecommunications Institute of Technology",

"Posts\*and\*Telecommunications\*Institute\*of\*Technology, Hanoi, VietNam");

cout << "\n Result: " << lenc;

cout << endl << "\n\n T h e E n d \n";

return 0;

}

#### Phương án 2. Dùng mảng 2D

**Algorithm**

d[i,j] = Len(i,j) call hàm Len(i,j) không quá 1 lần

Khởi trị mảng d?

Trật tự điền: trên xuống, dưới lên, trái phải?

Len(i,j) = max len A(0..i), B(0..j)

A(i) = B(j) : Len(i,j) = Len(i-1,j-1) + 1 // (trên, trái) + 1

A(i) != B(j): Len(i,j) = max(Len(i-1,j), len(i, j-1)) // max((trên,j). (i, trái))

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| B | A | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |
| **a** | **1** | **b** | **2** | **c** | **3** | **4** | **5** | **d** | **1** | **2** | **3** |  |  |  |  |  |  |  |  |  |  |
| **a** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |  | x |  | \* | + |  |  |  |  |
| **b** | 1 | 1 | 2 | 2 | \* | + | 2 | 2 | 2 | 2 | 2 | 2 |  |  | y |  | + | ? |  |  |  |  |
| **c** | 1 | 1 | 2 | 2 | + | ? |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **d** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Complexity**: len(a)len(b)

**Program**

/\*

Name: LCS.CPP (Ver. 2: fill aray 2D)

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description:

Longest sub string (Xau con chung dai nhat)

A: "Posts and Telecommunications Institute of Technology"

B: "Posts\*and\*Telecommunications\*Institute\*of\*Technology, Hanoi, VietNam"

C: "Posts and Telecommunications Institute of Technology"

Dynamic Programming

Len(i,j) = max len A(0..i), B(0..j)

A(i) = B(j) : Len(i,j) = Len(i-1,j-1) + 1

A(i) != B(j): Len(i,j) = max(Len(i-1,j), len(i, j-1))

------x

----------x

------x

----------y

\*/

#include <bits/stdc++.h>

using namespace std;

string a, b;

short \*\*d;

void Go() {

cout << " ? ";

fflush(stdin);

if(cin.get() == '.') exit(0);

}

void Print(short a[], short d, short c, const char \* msg = "") {

cout << msg;

for(short i = d; i <= c; ++i)

cout << " " << a[i];

}

void Print(short \*\*a, short d1, short c1,

short d2, short c2, const char \* msg = "") {

cout << msg;

for(short i = d1; i <= c1; ++i)

Print(a[i], d2, c2, "\n ");

}

// 2D

short Run(string inpa, string inpb) {

a = inpa; b = inpb;

short lena = a.length();

short lenb = b.length();

cout << "\n Input a: " << a << " lena = " << lena;;

cout << "\n Input b: " << b << " lenb = " << lenb;

d = new short \* [lena];

for(short i = 0; i < lena; ++i) {

d[i] = new short[lenb];

fill(d[i], d[i]+lenb, 0);

}

// dien dong 0

short i, j, c = 0;

i = 0;

for(j = 0; j < lenb; ++j) {

if (a[j] == b[i]) c = 1;

d[i][j] = c;

}

// Print(d[0], 0, lenb-1, "\n Row 0: ");

// dien cot 0

c = 0;

j = 0;

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

if (a[i] == b[j]) c = 1;

d[i][j] = c;

}

//Print(d, 0, lena-1, 0, lenb-1, "\n Init d: ");

// dien tu dong d[1]..

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

for(j = 1; j < lenb; ++j) {

d[i][j] = (a[i] == b[j]) ? d[i-1][j-1] + 1

: max(d[i-1][j], d[i][j-1]);

}

}

//Print(d, 0, lena-1, 0, lenb-1, "\n Final d: ");

return d[lena-1][lenb-1]; // Len(a.length()-1, b.length()-1);

}

main() {

short lenc = Run("abcd", "a1b2c345d123");

cout << "\n Result: " << lenc;

lenc = Run("Posts and Telecommunications Institute of Technology",

"Posts\*and\*Telecommunications\*Institute\*of\*Technology, Hanoi, VietNam");

cout << "\n Result: " << lenc;

// string s = "PostsandTelecommunicationsInstituteofTechnology";

// cout << "\n " << " len = " << s.length();

cout << endl << "\n\n T h e E n d \n";

return 0;

}

#### Phương án 3. Dùng 2 mảng 1D

/\*

Name: LCS.CPP (Ver. 3: fill aray 1D)

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description:

Longest sub string (Xau con chung dai nhat)

A: "Posts and Telecommunications Institute of Technology"

B: "Posts\*and\*Telecommunications\*Institute\*of\*Technology, Hanoi, VietNam"

C: "Posts and Telecommunications Institute of Technology"

Dynamic Programming

Len(i,j) = max len A(0..i), B(0..j)

A(i) = B(j) : Len(i,j) = Len(i-1,j-1) + 1

A(i) != B(j): Len(i,j) = max(Len(i-1,j), len(i, j-1))

------x

----------x

------x

----------y

\*/

#include <bits/stdc++.h>

using namespace std;

string a, b;

void Go() {

cout << " ? ";

fflush(stdin);

if(cin.get() == '.') exit(0);

}

void Print(short a[], short d, short c, const char \* msg = "") {

cout << msg;

for(short i = d; i <= c; ++i)

cout << " " << a[i];

}

short Run(string inpa, string inpb) {

a = inpa; b = inpb;

short lena = a.length();

short lenb = b.length();

cout << "\n Input a: " << a << " lena = " << lena;;

cout << "\n Input b: " << b << " lenb = " << lenb;

short \*x, \*y, \*z;

x = new short[lena]; // dong tren

y = new short[lena]; // dong duoi

// Init x; j : 0..lena a[j] ? b[0]

short i, j, c = 0;

for(j = 0; j < lena; ++j) {

if (a[j] == b[0]) c = 1;

x[j] = c;

}

Print(x, 0, lena-1, "\n Init x: ");

c = (b[0] == a[0]);

for(i = 1; i < lenb; ++i) { // dong i

// x -> y: cap nhat dong y qua x

if (b[i] == a[0]) c = 1;

y[0] = c;

for(j = 1; j < lena; ++j) {

y[j] = (a[j] == b[i]) ? x[j-1] + 1

: max(y[j-1], x[j]);

}

//Print(y, 0, lena-1, "\n y: "); Go();

z = x; x = y; y = z;

}

// Print(y, 0, lena-1, "\n Final y: ");

return x[lena-1];

}

main() {

short lenc = Run("a1b2c345d123", "abcd");

cout << "\n Result: " << lenc;

lenc = Run("Posts and Telecommunications Institute of Technology",

"Posts\*and\*Telecommunications\*Institute\*of\*Technology, Hanoi, VietNam");

cout << "\n Result: " << lenc;

string s = "PostsandTelecommunicationsInstituteofTechnology";

cout << "\n " << " len = " << s.length();

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### DÃY CON TĂNG DÀI NHẤT

**Algorithm**

Dynamic programming

Len(i) = chiều dài max của dãy con tăng khi duyệt a[0..i].

len(i) = max { len(j)+1: j < I , a[j] < a[i] }

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *i* | *0* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* | *11* | *12* | *13* | *14* | *15* |
| a | **2** | **1** | **7** | **3** | **3** | **1** | **5** | **2** | **1** | **7** | **2** | **9** | **4** | **11** | **13** | **3** |
| len | 1 | 1 | 2 | 2 | 2 | 1 | 3 | 2 | 1 | 4 | 2 | 5 | 3 | 6 | **7** | 3 |

return max(len);

**Complexity**: O(n)

**Program**

/\*

Name: INCSUB.CPP

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description:

Day con tang chát

\*/

#include <bits/stdc++.h>

using namespace std;

int a[] = {2, 1, 7, 3, 3, 1, 5, 2, 1, 7, 2, 9, 4, 11, 13, 3};

void Go() {

cout << " ? ";

fflush(stdin);

if(cin.get() == '.') exit(0);

}

void Print(int a[], int d, int c, const char \* msg = "") {

cout << msg;

for(int i = d; i <= c; ++i)

cout << " " << a[i];

}

// Day con tang chat dai nhat

int Run() {

int n = sizeof(a) / sizeof(int);

cout << "\n n = " << n;

int len[n];

Print(a, 0, n-1, "\n Input a: ");

fill(len, len+n, 1);

Print(len, 0, n-1, "\n Init len: ");

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

// for(int j = i-1; j >= 0; --j) {

for(int j = i-1; j >= len[i]; --j) {

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

len[i] = max(len[i], len[j]+1);

}

}

}

Print(len, 0, n-1, "\n Result len: ");

int maxlen = len[0];

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

maxlen = max(maxlen, len[i]);

}

cout << "\n Result: " << maxlen;

return maxlen;

}

main() {

Run();

cout << endl << "\n\n T h e E n d \n";

return 0;

}

**Note**

Dùng thêm con trỏ ngược có thể viết ra dãy con tăng dài nhất.

### DÃY CON CÓ TỔNG S

**Algorithm**

Xét hàm bool 2 ngôi d(i, t) = 1 nếu a[0..i] có chứa một dãy con có tổng t; ngược lại d(i, t) = 0.

Ta tính kết quả d(n,s).

Ta có

Khởi trị; Với dãy a[0..0] có 1 phần tử là a[0]: d(0, a[0]) = 1.

Với i : 2..n: xét dãy a[0..i] với t: 1..s

d(i, t) = d(i-1, t) or d(i-1,t-a[i])

Ví dụ

n = 8; s = 10.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *i* | *0* | *1* | *2* | *3* | *4* | *5* | *6* | *7* |
| a | **6** | **5** | **7** | **2** | **9** | **3** | **7** | **1** |
|  | x |  |  |  |  | x |  | x |
|  |  | x |  | x |  | x |  |  |
|  |  |  | x | x |  |  |  | x |
|  |  |  |  |  | x |  |  | x |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Duyệt ngược t = s 🡨 a[i] | | | | | | | | | | | Xét dãy a[0..i] | | | | | | | |
| t | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |  | *0* | *1* | *2* | *3* | *4* | *5* | *6* | *7* |
| *i = 0* | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |  | 6 |  |  |  |  |  |  |  |
| *1* | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |  | 6 | 5 |  |  |  |  |  |  |
| *2* | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |  | 6 | 5 | 7 |  |  |  |  |  |
| *3* | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |  | 6 | 5 | 7 | 2 |  |  |  |  |
| *4* | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |  | 6 | 5 | 7 | 2 | 9 |  |  |  |
| *5* | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |  | 6 | 5 | 7 | 2 | 9 | 3 |  |  |
| *6* | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |  | 6 | 5 | 7 | 2 | 9 | 3 | 7 |  |
| *7* | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  | 6 | 5 | 7 | 2 | 9 | 3 | 7 | 1 |

Có thể thêm Phase 1. bor đi những số > s.

**Complexity**: O(ns)

**Program**

/\*

Name: SUB.CPP

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description:

Day con co tong s

Dynamic programming

\*/

#include <bits/stdc++.h>

using namespace std;

int a[] = {6, 5, 7, 2, 9, 3, 7, 1};

void Go() {

cout << " ? ";

fflush(stdin);

if(cin.get() == '.') exit(0);

}

void Print(int a[], int d, int c, const char \* msg = "") {

cout << msg;

for(int i = d; i <= c; ++i)

cout << " " << a[i];

}

void Print(bool a[], int d, int c, const char \* msg = "") {

cout << msg;

for(int i = d; i <= c; ++i)

cout << " " << a[i];

}

void Run(int n, int s) {

//cout << "\n Run n = " << n << " s = " << s;

bool d[s+1];

fill(d, d+s+1, 0);

d[0] = 1;

//Print(d, 0, s, "\n Init d: ");

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

// duyet nguoc t = s..a[i]

for (int t = s; t >= a[i]; --t) {

d[t] = d[t] || d[t-a[i]];

} // t

if(d[s]) {

cout << " YES. Break at " << i;

return;

}

// Print(d, 0, s, " d: ");

} // i

if(d[s]) cout << " YES.";

else cout << " NO.";

}

main() {

int n = sizeof(a) / sizeof(int);

cout << "\n n = " << n;

Print(a, 0, n-1, "\n Input a: ");

for(int s = 0; s <= 45; ++s) {

cout << "\n Day con co tong = " << s << "? ";

Run(n, s); //Go();

}

cout << endl << "\n\n T h e E n d \n";

return 0;

}

**Bài tương tự**

Một người bán hàng có n tờ tiền mệnh giá a(i).

Người đó có thể trả lại s đồng cho khách hàng hay không?

### SỐ NHỎ NHẤT CÓ N ƯỚC SỐ

Ta cần giải một số bài liên quan đến ước số

### Hàm Tau

là hàm cho số ước của số nguyên dương n (kể cả 1 và n). Ví dụ ,

|  |  |
| --- | --- |
|  |  |
|  |  |

Nhận xét

* Các số chính phương, tức là các số dạng a\*a có số ước lẻ, còn lại là các số có số ước chẵn.
* Số nguyên tố là các số có đúng 2 ước.
* Nếu a và b nguyên tố cùng nhau thì , Ví dụ
* Nếu x được phân tích ra thừa số nguyên tố

thì

Ví dụ

**Algorithm**

Dựa vào tính đối xứng. Nếu x = ab và a ≠ b thì x có thêm hai ước là a và b; nếu a = b thì x có thêm 1 ước là a

// O(Sqrt(x))

int Tau(int x) {

int u, t = 2; // x co san 2 uoc la 1 va x

for(u = 2; u\*u <= x; ++u)

if(x % u == 0) t += 2;

--u;

return (u\*u == x) ? t-1 : t;

}

**Complexity**: O(sqr(x))

**Program**

/\*

Name: Tau.CPP

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description:

Ham Tau(x) = so uoc cua x

\*/

#include <bits/stdc++.h>

using namespace std;

void Go() {

cout << " ? ";

fflush(stdin);

if(cin.get() == '.') exit(0);

}

// O(Sqrt(x))

int Tau(int x) {

int t = 2, u;

for(u = 2; u\*u <= x; ++u)

if(x % u == 0) t += 2;

--u;

return (u\*u == x) ? t-1 : t;

}

main() {

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

cout << "\n Tau(" << i << ") = " << Tau(i);

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### All Tau

Liệt kê số ước của các số từ 1 đến n.

**Algorithm**

Phỏng theo thuật toán Sàng Eratosthenes.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| init | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 1 | 2 | 2 | **3** | 2 | **3** | 2 | **3** | 2 | **3** | 2 | **3** | 2 | **3** | 2 | **3** | 2 | **3** | 2 | **3** |
| 3 | 1 | 2 | 2 | 3 | 2 | **4** | 2 | 3 | **3** | 3 | 2 | **4** | 2 | 3 | **3** | 3 | 2 | **4** | 2 | 3 |
| 4 | 1 | 2 | 2 | 3 | 2 | 4 | 2 | **4** | 3 | 3 | 2 | **5** | 2 | 3 | 3 | **4** | 2 | 4 | 2 | **4** |
| 5 | 1 | 2 | 2 | 3 | 2 | 4 | 2 | 4 | 3 | **4** | 2 | 5 | 2 | 3 | **4** | 4 | 2 | 4 | 2 | **5** |
| 6 | 1 | 2 | 2 | 3 | 2 | 4 | 2 | 4 | 3 | 4 | 2 | **6** | 2 | 3 | 4 | 4 | 2 | **5** | 2 | 5 |
| 7 | 1 | 2 | 2 | 3 | 2 | 4 | 2 | 4 | 3 | 4 | 2 | 6 | 2 | **4** | 4 | 4 | 2 | 5 | 2 | 5 |
| 8 | 1 | 2 | 2 | 3 | 2 | 4 | 2 | 4 | 3 | 4 | 2 | 6 | 2 | 4 | 4 | **5** | 2 | 5 | 2 | 5 |
| 9 | 1 | 2 | 2 | 3 | 2 | 4 | 2 | 4 | 3 | 4 | 2 | 6 | 2 | 4 | 4 | 5 | 2 | **6** | 2 | 5 |
| 10 | 1 | 2 | 2 | 3 | 2 | 4 | 2 | 4 | 3 | 4 | 2 | 6 | 2 | 4 | 4 | 5 | 2 | 6 | 2 | **6** |

Khởi trị t[1] = 1; t[i] = 2, i = 2..n;

Lặp 2..n/2:

++t[j], j = 2i..kj ≤ n

**Complexity**: O(n^2)

**Program**

Tau(x), x = 1..200K.

/\*

Name: ALLTau.CPP

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description:

Tau(x), x : 1..n

\*/

#include <bits/stdc++.h>

using namespace std;

typedef vector<int> VI;

VI tau;

void Go() {

cout << " ? ";

fflush(stdin);

if(cin.get() == '.') exit(0);

}

// O(Sqrt(x))

int Tau(int x) {

int t = 2, u;

for(u = 2; u\*u <= x; ++u)

if(x % u == 0) t += 2;

--u;

return (u\*u == x) ? t-1 : t;

}

void Print(int t[], int d, int c, const char \* msg = "") {

cout << msg;

for(int i = d; i <= c; ++i)

cout << " " << t[i];

}

void Print(VI t, int d, int c, const char \* msg = "") {

cout << msg;

for(int i = d; i <= c; ++i)

cout << " " << t[i];

}

VI AllTau(int n) {

int t[n+1];

fill(t, t+n+1, 2);

t[1] = 1;

//Print(t, 1, n, "\n Init t: ");

int n2 = n/2;

for(int i = 2; i <= n2; ++i) {

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

++t[j];

}

//Print(t, 1, n, "\n Result t: ");

VI tau;

tau.push\_back(0); // t[0]

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

// cout << "\n Tau(" << i << ") = " << t[i];

tau.push\_back(t[i]);

}

return tau;

}

void Test() {

for(int i = 1; i < tau.size(); ++i) {

if (Tau(i) != tau[i]) {

cout << "\n Error at i = " << i << ": " << Tau(i) << " != " << tau[i];

return;

}

}

cout << "\n CORRECT.";

}

main() {

tau = AllTau(200000);

//Print(tau, 1, tau.size()-1, "\n Result: ");

Test();

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### Tau200K

**Algorithm**

Khảo sát các số 1..200K

* AllTau(1..200K)
* Find maxtau = 160: trong các số 1..200K thì có ước lớn nhất là 160
* Với các ước từ u = 1..160 lấy số đầu tiên n có tau(n) = u

Complexity

**Program**

/\*

Name: ALLTau.CPP

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description:

Tau(x), x : 1..n

\*/

#include <bits/stdc++.h>

using namespace std;

const int MN = 200000;

typedef vector<int> VI;

int maxtau;

void Go() {

cout << " ? ";

fflush(stdin);

if(cin.get() == '.') exit(0);

}

// O(Sqrt(x))

int Tau(int x) {

int t = 2, u;

for(u = 2; u\*u <= x; ++u)

if(x % u == 0) t += 2;

--u;

return (u\*u == x) ? t-1 : t;

}

void Print(int t[], int d, int c, const char \* msg = "") {

cout << msg;

for(int i = d; i <= c; ++i)

cout << " " << t[i];

}

void Print(VI t, int d, int c, const char \* msg = "") {

cout << msg;

for(int i = d; i <= c; ++i)

cout << " " << t[i];

}

// So nho nhat x co Tau = x

// xet cac so 1..n

VI Num(int n) {

int t[n+1];

fill(t, t+n+1, 2);

t[1] = 1;

//Print(t, 1, n, "\n Init t: ");

int n2 = n/2;

for(int i = 2; i <= n2; ++i) {

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

++t[j];

}

maxtau = 2;

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

maxtau = max(maxtau, t[i]);

}

cout << "\n maxtau = " << maxtau;

VI num;

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

num.push\_back(0);

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

int v = t[i];

if(num[v] == 0)

num[v] = i;

}

return num;

}

main() {

VI num = Num(MN);

for(int i = 1; i < num.size(); ++i) {

cout << "\n So nho nhat co " << i << " uoc la " << num[i];

}

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### TauN

Cho số nguyên dương N. Nhiệm vụ của bạn là tìm số K nhỏ nhất, sao cho K có đúng N ước số.

Đáp án sẽ là số không quá 18 chữ số thập phân (long long).

Hiểu đề

Ta quy ước rằng mọi số đề cập trong bài này đều là các số nguyên dương.

Trong toán học, hàm *τ*(n) (đọc là tau) cho ra *số ước của số* n (tính cả 1 và n).

Ví dụ,

τ(6) = 4, vì 6 có 4 ước là 1, 2, 3 và 6.

τ(12) = 6, vì 12 có 6 ước là 1, 2, 3, 4, 6 và 12.

Lưu ý rằng có nhiều số cùng có số ước như nhau, ví dụ, mọi số nguyên tố p đều có τ(p) = 2.

Đề bài yêu cầu: cho N, tìm min K để τ(K) = N.

Trước hết chúng ta làm quen với vài tính chất của hàm τ.

* T1. p là số nguyên tố khi và chỉ khi τ(p) = 2.
* T2. Nếu n là số *chính phương*, tức là số bằng bình phương của số a nào đó,

thì τ(n) là một số lẻ, ngoài ra, τ(n) là một số chẵn.

Ví dụ, τ(4) = 3; τ(9) = 3; τ(5) = 2; τ(15) = 4.

* T3. Số 1 là số duy nhất có τ(1) = 1; các số còn lại đều có τ(x) ≥ 2.
* T4. Hàm τ là hàm *nhân*, tức là nếu a và b là hai số nguyên tố cùng nau

(hai số có ước chung lớn nhất là 1, (a,b) = 1), thì τ(ab) = τ(a)τ(b).

Ví dụ, τ(15) = τ(3)τ(5) = 2⋅2 = 4.

* T5. Nếu *n* được phân tích ra thừa số nguyên tố,

thì

Ví dụ, .

Tính chất T5 cho biết tính không đơn trị của hàm τ: có nhiều số khác nhau cho cùng giá trị τ,

ví dụ, . Mọi số nguyên tố đều có cùng .

Đây là điểm khó cho bài này.

* T6. Nếu N là số nguyên tố thì theo tính chất T5, số K nhỏ nhất có τ(K) = N là

Ví dụ, N = 7, τ(64) = 7.

* T7. Cho số n, ta phân tích số n ra tích các thừa số nguyên tố không nhất thiết khác nhau nhưng được

sắp giảm dần:

Ví dụ, 12 = 3\*2\*2; 315 = 7\*5\*3\*3.

Năm 1968 Grost [2] đề xuất hàm A(n) là hàm cho ra *số nhỏ nhất có đúng n ước* và gọi số n là *số thường* nếu

trong đó 2, 3,...,q là dãy m số nguyên tố đầu tiên.

Grost đã thiết lập công thức tính A(n) cho các số n với m ≤ 6.

Ta cũng quy ước chấp nhạn 1 là số thường, các số không phải là số thường được gọi là *số trội.*

Năm 2006, Brown [1] chứng minh rằng

* *Mọi số căn bậc hai tự do đều là các số thường.*
* *Nếu n là số căn bậc hai tự do lẻ thì 4n cũng là số thường.*
* *Hầu hết các số đều là số thường.*

Số *căn bậc hai tự do* là những số trong dạng phân tích (\*) các số nguyên tố đều khác nhau đôi một.

Ví dụ,

* Các số nguyên tố là số *căn bậc hai tự do*
* Các số 15 = 5\*3, 42 = 7\*3\*2 là số *căn bậc hai tự do*
* *Số* 24 = 3⋅2⋅2⋅2 không phải là số *căn bậc hai tự do*

Theo tính chất T6, khi N nguyên tố thì .

* T8. Ta xét một số trường hợp ngoại lệ [1][2]
* (8.1) Nếu N là tích của 2 số nguyên tố pq được sắp giảm dần (không nhất thiết khác nhau), p ≥ q

thì .

Ví dụ

. Vậy 36 là số nhỏ nhất có 9 ước: τ(36) = 9.

. Vậy 144 là số nhỏ nhất có 15 ước: τ(144) = 15.

* (8.2) Nếu N là tích của 3 số nguyên tố pqr được sắp giảm dần (không nhất thiết khác nhau) N = pqr, p ≥ q ≥ r thì , ngoại trừ một trường hợp (\*) sau đây:
* N = 8 = 2⋅2⋅2 đáng lẽ A(8) = 2⋅3⋅5 = 30 nhưng chỉ cần K = 24. Vậy 24 là số nhỏ nhất có 8 ước: τ(24) = 8 (\*).
* (8.3) Nếu N là tích của 4 số nguyên tố pqrs được sắp giảm dần (không nhất thiết khác nhau) N = pqrs, p ≥ q ≥ r ≥ s thì , ngoại trừ hai trường hợp (\*) và (\*\*) sau đây:
* N = 16 = 2⋅2⋅2⋅2, đáng lẽ A(16) = 2⋅3⋅5⋅7 = 210 , nhưng chỉ cần = 120. Vậy 120 là số nhỏ nhất có 16 ước: τ(120) = 16 (\*).
* N = 24 = 3⋅2⋅2⋅2 = 24, đáng lẽ nhưng chỉ cần . Vậy 360 là số nhỏ nhất có 24 ước: τ(360) = 24 (\*\*).
* (8.4) Nếu N là tích của 5 hoặc 6 số nguyên tố không nhất thiết khác nhau thì có nhiều ngoại lệ. Trường hợp này chúng ta phải Test, cụ thể là sau khi tìm được giá trị K, bạn phải giảm dần K để tìm Kmin. Bạn đọc có thể tham khảo [1].

Ví dụ

N = 32 = 2⋅2⋅2⋅2⋅2, đáng lẽ A(N) = 2⋅3⋅5⋅7⋅11 = 2310, nhưng chỉ cần K = 2⋅2⋅2⋅3⋅5⋅7 = 840. Vậy 840 là số nhỏ nhất có 32 ước: τ(840) = 32.

**Tham khảo**

[1] R. Brown, The minimal number with a given number of divisors, Journal of Number Theory 116 (2006) 150-158.

[2] M. E. Grost. The Smallest Number with a Given Number of Divisors. The American Mathematical Monthly, September 1968, pp. 725-729.

Thuật toán khi đó sẽ như sau:

Input n

Output Số k nhỏ nhất: Tau(k) = n.

Method

* Xét các trương hợp đặc biệt
  + n nguyên tố:
  + n = 8: *return* 24;
  + n = 16: *return* 120;
  + n = 24: *return* 360;
  + n = 32: *return* 840;
* Phân tích n ra thừa số p[1..k] giảm dần.
* if k < 5 or n là căn bậc hai tự do
* nếu không: kiểm tra thêm

**Complexity**: chi phí phân tích ra thừa số nguyên tố O()

**Program**

/\*

Name: TauN.CPP

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description:

So nho k nhat Tau(k) = n

Xet cac truong hop dac biet:

n nguyen to: return 2^(n-1)

n = 8: return 24;

n = 16: return 120;

n = 24: return 360;

n = 32: return 840;

Phan tuch n ra thua so ng to giam dan p[1..k].

if (k < 5 || n is ordinary) return product (q[i]^(p[i]-1)),i = 1..k

else ki?m tra thêm

\*/

#include <bits/stdc++.h>

using namespace std;

int q[] = {1, 2, 3, 5, 7, 11, 13}; // cac so ng to lien tiep

int \*p;

typedef unsigned long long UL;

typedef vector<int> VI;

void Go() {

cout << " ? ";

fflush(stdin);

if(cin.get() == '.') exit(0);

}

void Print(int t[], int d, int c, const char \* msg = "") {

cout << msg;

for(int i = d; i <= c; ++i)

cout << " " << t[i];

}

void Print(VI v, const char \* msg = "") {

cout << msg;

for(int i = 1; i < v.size(); ++i)

cout << " " << v[i];

}

// O(Sqrt(x))

int Tau(int x) {

int t = 2, u;

for(u = 2; u\*u <= x; ++u)

if(x % u == 0) t += 2;

--u;

return (u\*u == x) ? t-1 : t;

}

// phan tich n thanh p[1..m] sap giam

int Decomp(int n) {

cout << "\n Decomposite " << n;

VI v;

v.push\_back(1); // bo qua v[0]

while(n % 2 == 0) {

v.push\_back(2);

n /= 2;

}

for (int u = 3; u\*u <= n; u += 2) {

while(n % u == 0) {

v.push\_back(u);

n /= u;

}

}

if (n > 1) v.push\_back(n);

//Print(v, "\n Decom: ");

int m = v.size()-1;

cout << " m = " << m;

// ghi nguoc vao p

p = new int[m+1]; // bo qua p[0] con p[1..m]

int j = 1;

for (int i = m; i > 0; --i, ++j)

p[j] = v[i];

Print(p, 1, m, "\n p: ");

return m;

}

inline UL Product(int m) {

UL z = 1L;

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

z \*= (UL)pow(q[i],p[i]-1);

return z;

}

// y: z-1..1, Tau(x) < z

UL Test(int n, UL z) {

cout << "\n Test n = " << n << " z = " << z; Go();

UL y, ymin = z, one = 1L;

for (y = z-1; y > one; --y)

if (Tau(y) == n) ymin = y;

// cout << " Test = " << n << " " << k << " -> " << ymin;

return ymin;

}

// Kiem tra p[1..k] co trung lap?

bool SquareFree(int m) {

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

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

if(p[j] == p[i]) return false;

}

}

return true;

}

UL A(int n) {

//cout << "\n n = " << n;

switch(n) {

case 1: return 1;

case 8: return 24;

case 16: return 120;

case 24: return 360;

case 32: return 840;

} // switch

int m = Decomp(n); // n = p[1..m]

cout << "\n n = " << n << " m = " << m;

UL z = Product(m);

if (m < 5) return z;

if (SquareFree(m)) return z;

// Kiem tra them

return Test(n,z);

}

void Run() {

UL k;

for(int N = 1; N <= 60; ++N) {

k = A(N);

cout << "\n So nho nhat co " << N << " uoc la k = " << k

<< " ? Tau(" << k << ") = " << Tau(k);

}

}

main() {

Run();

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### ĐƯỜNG ĐI VỚI CHI PHÍ NHỎ NHẤT

**Algorithm**

Dijkstra

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 | 2 | 3 |  | 0 | ∞ | ∞ |  | **1** | 3 | ∞ |  | 1 | **3** | ∞ |  | 1 | 3 | 6 |  | 1 | 3 | 6 |
|  | 4 | 8 | 2 |  | ∞ | ∞ | ∞ |  | 5 | 9 | ∞ |  | 5 | 9 | 5 |  | **5** | 9 | 5 |  | 5 | 9 | **5** |
|  | 1 | 5 | 3 |  | ∞ | ∞ | ∞ |  | ∞ | ∞ | ∞ |  | ∞ | ∞ | ∞ |  | 6 | 10 | ∞ |  | 6 | 10 | 8 |
|  |  |  |  |  | Khởi trị | | |  | min = (1, 1) | | |  | min = (1, 2) | | |  | min = (2, 1) | | |  | min = (2, 3) | | |
|  | 1 | 3 | **6** |  | 1 | 3 | 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 5 | 9 | 5 |  | 5 | 9 | 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 6 | 10 | 8 |  | **6** | 10 | 8 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | min = (1, 3) | | |  | min = (3, 3) | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Complexity**:

**Program**

/\*

Name: SHORTESTPATH.CPP

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description:

DUONG CO CHI PHI NHO NHAT TU (1,1) DEN (n,m)

Dijkstra: Dynamic programming

\*/

#include <bits/stdc++.h>

using namespace std;

const char \* fn = "SHORTESTPATH.INP";

int n, m;

int \*\*a;

int \*\*p;

bool \*\*rest;

int inf; // vo cung

int imin, jmin;

void Go() {

cout << " ? ";

fflush(stdin);

if(cin.get() == '.') exit(0);

}

void Print(int \*a, int d, int c, const char \* msg = "") {

cout << msg;

for(int i = d; i <= c; ++i)

cout << " " << a[i];

}

void Print(int \*\*a, int d1, int c1,

int d2, int c2, const char \* msg = "") {

cout << msg;

for(int i = d2; i <= c2; ++i)

Print(a[i], d1, c1, "\n ");

}

void Print(bool \*a, int d, int c, const char \* msg = "") {

cout << msg;

for(int i = d; i <= c; ++i)

cout << " " << a[i];

}

void Print(bool \*\*a, int d1, int c1,

int d2, int c2, const char \* msg = "") {

cout << msg;

for(int i = d2; i <= c2; ++i)

Print(a[i], d1, c1, "\n ");

}

void ReadInput() {

ifstream f(fn);

if(f.fail()) {

cerr << "\n No input file " << fn;

exit(1);

}

f >> n >> m;

cout << "\n row = " << n << " column = " << m;

a = new int\*[n];

p = new int\*[n];

rest = new bool\*[n];

inf = 10;

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

a[i] = new int[m];

p[i] = new int[m];

rest[i] = new bool[m];

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

f >> a[i][j];

inf += a[i][j];

}

}

Print(a, 0, n-1, 0, m-1, "\n Init a: ");

cout << "\n vo cung = " << inf;

}

void GetMin() {

int valmin = inf;

//cout << "\n Init valmin: " << valmin; //Go();

imin = 0, jmin = 0;

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

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

if(rest[i][j]) {

if(p[i][j] < valmin) {

valmin = p[i][j];

//cout << "\n valmin: " << valmin; //Go();

imin = i; jmin = j;

//cout << "\n imin: " << imin << " : jmin = " << jmin; // Go();

} // if

} // if

} // j

} // i

// cout << "\n imin = " << imin << " : " << " jmin = " << jmin; Go();

}

void Update() {

int x = p[imin][jmin];

int i, j;

// qua trai

if(jmin > 0) {

j = jmin-1;

if(rest[imin][j]) {

if(x+a[imin][j] < p[imin][j])

p[imin][j] = x+a[imin][j];

}

}

// qua phai

if(jmin < m-1) {

j = jmin+1;

if(rest[imin][j]) {

if(x+a[imin][j] < p[imin][j])

p[imin][j] = x+a[imin][j];

}

}

if(imin == n-1) return;

// co dong duoi

i = imin + 1;

// cheo trai

if(jmin > 0) {

j = jmin-1;

if(rest[i][j]) {

if(x+a[i][j] < p[i][j])

p[i][j] = x+a[i][j];

}

}

// cheo phai

if(jmin == m-1) return;

j = jmin+1;

if(rest[i][j]) {

if(x+a[i][j] < p[i][j])

p[i][j] = x+a[i][j];

}

}

void Dijkstra() {

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

fill(p[i], p[i]+m, inf);

fill(rest[i], rest[i]+m, 1);

}

p[0][0] = 0;

Print(p, 0, n-1, 0, m-1, "\n Init p: ");

Print(rest, 0, n-1, 0, m-1, "\n Init rest: ");

int nm = n\*m;

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

GetMin();

cout << "\n GetMin " << imin << " : " << jmin;

p[imin][jmin] = a[imin][jmin];

rest[imin][jmin] = 0;

Update();

}

cout << "\n Result: " << p[n-1][m-1];

}

void Run() {

ReadInput();

Dijkstra();

}

main() {

Run();

cout << endl << "\n\n T h e E n d \n";

return 0;

}

29 June 2023