**QUY HOẠCH ĐỘNG**

Mục luc

[I. Lý thuyết 2](#_Toc485544539)

[1. Giới thiệu: 2](#_Toc485544540)

[2. Một số khái niệm và các bước cài đặt: 2](#_Toc485544541)

[a. Các khái niệm 2](#_Toc485544542)

[b. Yêu cầu 2](#_Toc485544543)

[c. Các bước cài đặt một bài toán bằng quy hoạch động: 3](#_Toc485544544)

[II.Bài tập 4](#_Toc485544545)

[1. Dạng 1: Bài toán dãy con đơn diệu tăng dài nhất 4](#_Toc485544546)

[1.1. Bài toán: Cho thuê máy 4](#_Toc485544547)

[1.2. Bài toán: Bố trí phòng họp 4](#_Toc485544548)

[1.3. Bài toán: Dãy tam giác bao nhau 4](#_Toc485544549)

[1.4. Bài toán: Dãy đổi dấu 4](#_Toc485544550)

[1.5. Bài toán: Dãy số WAVIO 4](#_Toc485544551)

[2. Dạng 2: Bài toán Balo dạng 1 4](#_Toc485544552)

[2.1. Bài toán balo dạng 1 4](#_Toc485544553)

[2.2. Bài toán dãy con có tổng bằng S 4](#_Toc485544554)

[2.3. Bài toán chia kẹo 5](#_Toc485544555)

[2.4. Bài toán Market (Olympic Balkan 2000) 6](#_Toc485544556)

[2.5. Bài toán điền dấu biểu thức 6](#_Toc485544557)

[2.6. Bài toán Expression (ACM 10690) 6](#_Toc485544558)

[3. Dạng 3: Bài toán balo dạng 2 6](#_Toc485544559)

[3.1. Bài tóa balo dạng 2 6](#_Toc485544560)

[3.2. Bài toán Farmer (IOI 2004) 6](#_Toc485544561)

[3.3. Bài toán đổi tiền 6](#_Toc485544562)

[4. Dạng 4: Bài toán biến đổi xâu 6](#_Toc485544563)

[4.1. Bài toán biến đổi xâu 6](#_Toc485544564)

[4.2. Bài toán xâu con chung dài nhất 6](#_Toc485544565)

[4.3. Bài toán bắc cầu 6](#_Toc485544566)

[4.4. Bài toán PalinDrom (IOI 2000) 6](#_Toc485544567)

[5. Dạng 5: Bài toán nhân ma trận 6](#_Toc485544568)

[5.1. Bài toán nhân ma trận 6](#_Toc485544569)

[5.2. Bài toán chia đa giác 6](#_Toc485544570)

[5.3. Bài toán biểu thức số học (IOI 1999) 6](#_Toc485544571)

[6. Dạng 6: Bài toán ghép cặp 6](#_Toc485544572)

[6.1. Bài toán ghép cặp 6](#_Toc485544573)

[6.2. Bài toán câu lạc bộ 6](#_Toc485544574)

[6.3. Bài toán mua giày 6](#_Toc485544575)

[7. Dạng 7: Bài toán di chuyển 6](#_Toc485544576)

[7.1. Bài toán di chuyển 6](#_Toc485544577)

[7.2. Bài toán tam giác (IOI 1994) 6](#_Toc485544578)

[7.3. Bài toán con kiến 6](#_Toc485544579)

[--Hết-- 7](#_Toc485544580)

1. **Lý thuyết**
2. **Giới thiệu:**

Phương pháp quy hoạch động được nhà toán học Richard Bellman phát minh năm 1953. Quy hoạch động là kỹ thuật giải các bài toán có bản chất đệ quy nhằm tìm ra kết quả tối ưu cho bài toán trên tinh thần chia để trị, tức chia bài toán lớn thành nhiều bài toán nhỏ, giải tất cả các bài toán nhỏ để tim kết quả của bài toán lớn. Ý tưởng của nguyên lý tối ưu Bellman: “Với mỗi quá trình điều khiển tối ưu, đối với trạng thái bắt đầu A0, với trạng thái A trong quá trình đó, phần quá trình kể từ trạng thái A được xem như trạng thái bắt đầu cũng là tối ưu.” Hay nói cách là “nêu một cấu hình là tối ưu thì mọi cấu hình của nó đều tối ưu”.

1. **Một số khái niệm và các bước cài đặt:**
   1. **Các khái niệm**
      * ***Bài toán quy hoạch đông***: là bài toán giải theo phương pháp quy hoạch.
      * ***Công thức truy hồi***: là công thức phối hợp nghiệm của các bài toán con để có nghiệm bài toán lớn.
      * ***Cơ sở quy hoạch*** (cơ sở): là tập các bài toán nhỏ nhất có ngay lời giải.
      * ***Bảng phương án***: là không gian lưu trữ các kết quả của các bài toán con.
   2. **Yêu cầu**

Một bài toán có thể giải bằng quy hoạch động phải đảm bảo các yếu tố sau:

* + - Bài toán lớn phải phân rã được thành nhiều bài toán con mà sự kết hợp kết quả của các bài toán con cho ra được kết quả bài toán lớn.
    - Đảm bảo đủ không gian vật lí cho tất cả các bài toán con.
    - Quá trình từ bài toán con cho ra kết quả bài toán lớn phải là hữu hạn các bước.
  1. **Các bước cài đặt một bài toán bằng quy hoạch động:**
     + Phân tích bài toán và lập công thức truy hồi.
     + Giải tất cả bài toán cơ sở lưu vào bảng phương án (mảng 1, 2 chiều).
     + Dùng công thức truy hồi phối hợp những lời giải của bài toán nhỏ đã lưu trong bảng phương án để tìm lời giải bài toán lớn hơn và lưu vào bảng phương án. Lặp lại cho tới khi tìm được lời giải cho bài toán ban đầu.
     + Dựa vào bảng phương án ta truy ngược để tìm nghiệm tối ưu.

# **II.Bài tập**

## **Dạng 1: Bài toán dãy con đơn diệu tăng dài nhất**

### Bài toán: Cho thuê máy

Trung tâm tính toán hiệu năng nhận được đơn đặt hang của n khách hàng. Khách hàng i muốn sử dụng máy trong khoảng thời gian từ Si đến Fi và trả tiền thuê là Ci. Hãy bố trí lịch thuê máy để tổng số tiền thu được là lớn nhất mà thời gian sử dụng máy của hai khách bất kì được phục vụ không bằng nhau.

INPUT File

Dòng 1: N (số khách)

N dòng tiếp tiếp theo mỗi dòng có cấu trúc: Si Fi Ci

OUTPUT File:

Dòng 1: tổng số tiền (k khách)

K dòng tiếp theo mỗi dòng: Si Fi Ci

Phân tích bài toán:

Đầu tiên ta sắp xếp n đơn đặt hàng theo thứ tự tăng dần thời gian Fi. Ta sẽ thêm 2 phần tử A[0] và A[n+1] vào đơn đặt hàng với A[0] = {MIN,MIN,0}, A[n+1] = {MAX,MAX,0}. Có nghĩa là mọi đơn đặt hàng đều phải xếp sau đơn A[0] và trước đơn A[n+1] và dĩ nhiên đơn A[0], A[n+1] được miễn phí(Ci=0). Đặt Sum(i) là tổng số tiền thu được khi cho thuê các đơn hàng từ A[i] trở đi(sau khi đã sắp xếp). Như vậy bài toán sẽ chọn ra dãy các đơn hàng thỏa:

Ràng buộc: Không giao nhau, tức sau sắp xếp thì để xếp A[i] vào giữa A[k] và A[k+1] thì phải đảm bảo: A[i].s>=A[k].f và A[i].f<=A[k+1].s

Kết quả bài toán sẽ là max{Sum(i),1<i<n+1} ,(=Sum(0))

Cơ sở quy hoạch: trường hợp không cho thuê đơn nào thì kết quả Sum(n+1) = 0.

Công thức truy hồi:

Để tính được max(Sum(i)) thì ta cần tính mỗi Sum(i) với I chạy từ n->0. Giá trị Sum(i) được tính theo các Sum(i+1)… Sum(n+1) đã biết như sau:

Việc chọn A[i] để cho thuê là chính là thêm A[i] vào dãy các đơn đã cho thuê bắt đầu từ A[j] (j>i) sao cho:

+ A[i].f <= A[j].s

+Sum(j) = max với i<j<=n+1,

Vậy ta có công thức:

Tính Bảng phương án và truy vết:

Với mỗi Sum(i) = Ci+Sum(j) ta đặt truy vết Trace[i] = j để lưu lại rằng next(i) = j (đơn hàng phía sau đơn A[i] là đơn A[j]). Từ đó ta lần ngược lại theo mảng Trace sẽ được kết quả.

Ví dụ:

|  |  |
| --- | --- |
| **Input file** | **Output file** |
| 5  1 3 10  5 6 5  3 5 10  4 7 30  6 8 10 | 40  1 3 10  4 7 30 |

**Bảng phương án:**

**Sum**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| I | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| S[i],F[i] | MIN, MIN | 1, 3 | 3, 5 | 5, 6 | 4, 7 | 6, 8 | MAX, MAX |
| C[i] | 0 | 10 | 10 | 5 | 30 | 10 | 0 |
| Sum[i] | **40** | 40 | 25 | 15 | 30 | 10 | 0 |
| Trace[i] | 1 | 4 | 3 | 5 | 6 | 6 |  |

**Trace**

Từ bảng phương án ta có 2 lựa chọn:

+ Lựa chọn 1: (1,3) - (4,7) 🡪 tổng tiền: 40

+ Lựa chọn 2: (3, 5) - (5, 6) - (6, 8) 🡪 tổng tiền: 25

Chọn lựa chọn 1

***\*Code bài cho thuê máy:<thư mực source code>***

### Bài toán: Bố trí phòng họp

### Bài toán: Dãy tam giác bao nhau

### Bài toán: Dãy đổi dấu

### Bài toán: Dãy số WAVIO

## **Dạng 2: Bài toán Balo dạng 1**

### Bài toán balo dạng 1

Có n món đồ, vật thứ I có trọng lượng là wi và giá trị vi. Hãy chọn ra các món có thể bỏ vào một cái balo có trọng lượng tối đa là W sao cho tổng giá trị các món đồ là lớn nhất.

\* Đầu vào: Tập tin BALO.INP

- Dong 1: Chứa n W

- n dòng tiếp theo, dòng I chưa wi và ai

\* Đầu ra: Tập tin BALO.OUT

- Dong 1: Giá trị lớn nhất trong balo

- Dòng 2: Chỉ số những đồ vật được lấy

**\*Cách giải:**

Gọi ***F(i,j)*** là giá trị lớn nhất có thể có bằng cách chọn món đồ thứ {1, 2, 3, …, i} với giới hạn trọng lượng ***j***. Như vậy **F(n,W)** chính là giá trị lớn nhất khi chọn trong số n gói với giới hạn trọng lượng W.

1. **Công thức truy hồi:**

Với giới hạn trọng lượng ***j***, việc chọn phương án tối ưu trong số các gói {1, 2, 3, …,i-1, i} để có giá trị lớn nhất sẽ xảy ra 2 trường hợp:

* + *Không* chọn gói i: F(i ,j) = F(i-1, j)
  + *Chọn* gói i (wi <= j): F(i ,j) = vi + F(i-1,j - wi)
* Do đó muốn F(i,j) lớn nhất thì F(i,j) phải là số lớn nhất trong 2 trường hợp trên.

1. **Cơ sở quy hoạch đông:**

F(0,j) = 0 với mọi j vì giá trị lớn nhất bang 0 khi chọ trong số 0 món đồ.

1. **Bảng phương án:**

Bảng phương án F gồm n+1 dòng, W+1 cột. Đầu tiên cơ sở quy hoạch động tương đương dòng 0 gồm toàn số 0. Sử dụng công thức truy hồi dùng dòng i-1 để tính dòng i (i<=n)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **F** | **0** | **1** | **2** | **…** | **W** |
| **0** | 0 | 0 | 0 | 0 | 0 |
| **1** |  |  |  |  |  |
| **2** |  |  |  |  |  |
| **…** | … | … | … | … | … |
| **n** |  |  |  |  | **x** |

1. **Truy vết:**

Sau khi tính xong bảng phương án ta sẽ qua tâm đến F(n,W) = x vì đó chính là giá trị lớn nhất thư được trong cả n gói với giới hạn W.   
- Nếu F[n,W] = F[n-1,W] thì không chọn vật thứ n => Truy tiếp F[n-1,W]  
- Nếu F[n,W] != F[n-1,W] thì phép chọn có vật thứ n => Truy tiếp F[n-1,W-wn]  
Cứ tiếp tục cho tới khi truy tới dòng 0.

***\*Code bài toán balo dạng 1: <thư mục source code>***

### Bài toán dãy con có tổng bằng S

* + **Mô tả bài toán**: Cho dãy a1, a2, …,an. Tìm một dãy con của dãy đó có tổng bằng S.  
    (Dãy con không nhất thiết là liên tục)
    - Input:   
      - Dòng 1: N S  
      - Dòng 2: N phần từ của dãy
    - Output:  
      - Dòng 1: M S (M là số phần tử dãy con)  
      - Dòng 2: Dãy con thỏa yêu cầu (không có kết quả in “NO RESULT”).  
      Ví dụ:

|  |  |  |
| --- | --- | --- |
| Testcase | Input | Output |
| Test 01 | 8 10  3 2 5 4 2 4 7 9 | 3 10  3 2 5 |
| Test 02 | 10 20  1 5 3 6 -8 -5 7 4 12 -1 | 5 20  1 5 3 7 4 |
| Test 03 | 10 36  1 5 3 6 -8 -5 7 4 12 -1 | 0 36  NO RESULT |

* + **Ý tưởng giải:**   
    Đặt L(i,t) = 1 nếu có thể tạo tổng t từ các phần tử (a0->ai), ngược lại L(i,t)=0. Khi đó kết quả L(n,S) = 1 sẽ là kết quả của bài toán.
  + **Cấu trúc quy hoạch động:**
    - **Cơ sở quy hoạch:**   
      L(0 , 0)=1 dòng 0 không chứa phần tử nào của dãy ban đầu, có nghĩa là phép không chọn phần tử nào để có tổng S=0 là đúng. Các phần tử còn lại của dòng 0 đề bằng 0 (giải nghĩa: không chọn thì làm sao mà có đc tổng dương?). (Xem hình bên dưới)
    - **Công thức truy hồi:**

L(i,t)=1 ⬄ L(i-1,t) == 1 *or* L(i-1,t-ai) == 1  
,với 0<= t-ai <=S

* + - **Bảng phương án**:  
      Dùng mảng 2 chiều n+1 dòng, S+1 cột để lưu kết quả nhị phân (0,1)

(minh họa sau)

* + - **Truy vết:**  
      - Nếu L(n,S)==0 => không có kết quả.  
      - Nếu L(n,S)==1. Xét tiếp nếu L(n,S)==L(n-1,S) thì không chọn phần tử an. Xét tiếp đến L(n-1,S), nếu L(n-1,S) != L(n-2,S) thì chọn an-1, xét tiếp L(n—2,S-an-1). Cứ tiếp tục như vậy cho đến khi n==0 thì dừng.
    - **Chương trình cài đặt (demo test)**  
      - Testcase: N= 10, S=20, A={1 5 3 6 -8 -5 7 4 12 -1}

**Cơ sở quy hoạch**

- Bảng phương án và minh họa truy vết:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **i/j** | **A** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| 0 | **-** | **1** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | **1** | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | **5** | 1 | 1 |  |  |  | 1 | 1 |  |  | 0 |  |  |  |  |  |  |  |  |  |  |  |
| 3 | **3** | 1 | 1 |  | 1 | 1 | 1 | 1 |  | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |
| 4 | 6 | 1 | 1 |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  | 1 | 1 |  |  |  |  |  |
| 5 | -8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  | 1 | 1 |  |  |  |  |  |
| 6 | -5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  | 1 | 1 | 0 |  |  |  |  |
| 7 | **7** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 8 | **4** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 12 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 10 | -1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | **1** |

* Kết quả: **{1, 5, 3, 7, 4}.**

**\*Chú ý:** *Trong chương trình cài đặt phải tuân thủ mảng A vẫn chỉ số chạy từ 0, Ma trận L có dòng 0 là dòng cơ sở, n dòng tiếp theo ứng với n phần tử của A.*

* Cải tiến:  
  - Nhìn vào bản ma trận kết quả bên trên ta có nhận xét rằng tại dòng thứ 8 ta đã chọn được L(8, S)=1 rồi thì các phần tử bên dưới không cần xét.  
  - Vì kết quả dòng i chỉ phụ thuộc vào dòng i-1 hoàn toàn có thể dùng mảng 1 chiều để lưu kết quả này.

### Bài toán chia kẹo

* + Mô tả bài toán:  
    Cho n gói kẹo, gói thứ i có ai viên. Hãy chia các gói kẹo thành hai phần sao cho chênh lệch giữa 2 phần là ít nhất.
  + Ý tưởng giải  
    Gọi T là tổng số kẹo trong các gói. Chúng ta cần tìm số S lớn nhất sao cho:
    - Có một dãy con **a** (chọn từ n gói) có tổng bằng S.

Khi đó sẽ có cách chia tối ưu với chênh lệch nhỏ nhất (2T-S). Phần thứ nhất là dãu con **a** và phần thứ 2 là các gói còn lại sau khi chọn **a.** Cách tìm dãy con a hoàn toàn giống bài toán tìm dãy con có tổng bằng S bên trên, có chút biến đổi là bài toán chắc chắn có đáp án, có nghĩa là chọn S sao chọn S gần T/2 nhất và L(n,S)==1.

* + Cấu trúc quy hoach động
    - Cơ sở quy hoạch: (*Giống bài tìm dãy con có tổng bằng S phía trên).*
    - Công thức truy hồi: (*Giống bài tìm dãy con có tổng bằng S phía trên).*
    - Bảng phương án:   
      Tương tự bài toán tìm dãy con có tổng bằng S, ta dùng ma trận n+1 dòng T+1 cột nhị phân để lưu vết kết quả.
    - Truy vết:  
      - Sau khi xây dựng ma trận phương án, ta tiến hành chọn S trước, sau đó lần ngược lên trên để lấy các phần tử của dãy con **a.**   
      - Thuật toán lấy S:  
      - Lưu kết quả mỗi phần vào 2 mảng phụ

S=(int)T/2

While(L(S,n)!=1)

S=S-1

End while

* + Chương trình: xem trong thư mục source code  
    - Xét Dãy có N=6, A = {3 2 1 1 1 1}  
    - Bảng Phương án:  
    T= Sum(A) = 9 => ma trận 7x10

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **i/j** | **A** | **0** | **1** | **2** | **3** | **4** | **~~5~~** | **~~6~~** | **~~7~~** | **~~8~~** | **~~9~~** |
| **0** | - | 1 |  |  |  |  |  |  |  |  |  |
| **1** | 3 | 1 |  |  | 1 |  |  |  |  |  |  |
| **2** | 2 | 1 |  | 1 | 1 |  | ~~1~~ |  |  |  |  |
| **3** | 1 | 1 | 1 | 1 | 1 | 1 | ~~1~~ | ~~1~~ |  |  |  |
| **4** | 1 | 1 | 1 | 1 | 1 | 1 | ~~1~~ | ~~1~~ | ~~1~~ |  |  |
| **5** | 1 | 1 | 1 | 1 | 1 | 1 | ~~1~~ | ~~1~~ | ~~1~~ | ~~1~~ |  |
| **6** | 1 | 1 | 1 | 1 | 1 | 1 | ~~1~~ | ~~1~~ | ~~1~~ | ~~1~~ | ~~1~~ |

* Chọn: S=T/2 = 9/2=4, thỏa mãn L[n][S]==1
* Độ chênh lệch = T-2S = 9-2\*8=1
* Lần ngược từ L[n][S] ta được kết quả 2 phần cần chia:  
  + L[n][S] != L[n-1][S] => Chọn an vào phần 1 (màu vàng) = {a3, a1}
* + L[n][S] == L[n-1][S] => Chọn an vào phần 2 (màu đỏ) = {a6, a5, a4, a2}

### Bài toán Market (Olympic Balkan 2000)

* + **Mô tả bài toán:**

Người đánhcác Clememt bắt được n con cá, khối lượng mỗi con là ai (kg), đem ra bán ngoài chợ. Ở chợ người ta không mua cá theo từng kí lô mà mua theo khối lượng kí lô tron, chẳng hạn như 3 kg, 5 kg,… Chẳng hạn nếu có ba con cá với khối lượng lần lượt là 3 kg, 2 kg và 4 kg, một ví dụ như sau:  
- Mua 6 kg sẽ lấy con cá thứ hai và thứ ba.  
- Mua 3 kg sẽ lấy con cá thứ nhất.  
- Không thể mua 8 kg.  
🕙 **Câu hỏi:** *Nếu bạn là người mua cá đầu tiên thì có bao nhiêu trọng lượng khác nhau bạn có thể chọn?*

* + **Ý tưởng giải:**

Thực chất bài này giống y hệt bài dãy con tăng có tổng bằng S và bài chia kẹo. Điểm khác biệt là bây giờ ta đi đếm số tổng S có thể tạo được từ n phần tử. Hay nói cách khác là ta đi đếm số L[n,S]==1 và truy ngược lên

* + **Cấu trúc quy hoạch động:**
* Hoàn toàn giống 2 bài trên
* Thuật toán đếm số phương án mua cá:

S = T

While(S>0)

If(L(S,n)==1)

Count++

Truy ngược tìm nghiệm

End if

S=S-1

End while

* + **Chương trình:** <xem phần source code>

### Bài toán điền dấu biểu thức

### Bài toán Expression (ACM 10690)

## Dạng 3: Bài toán balo dạng 2

### Bài tóa balo dạng 2

### Bài toán Farmer (IOI 2004)

### Bài toán đổi tiền

## Dạng 4: Bài toán biến đổi xâu

### Bài toán biến đổi xâu

### Bài toán xâu con chung dài nhất

### Bài toán bắc cầu

### Bài toán PalinDrom (IOI 2000)

## Dạng 5: Bài toán nhân ma trận

### Bài toán nhân ma trận

### Bài toán chia đa giác

### Bài toán biểu thức số học (IOI 1999)

## Dạng 6: Bài toán ghép cặp

### Bài toán ghép cặp

### Bài toán câu lạc bộ

### Bài toán mua giày

## Dạng 7: Bài toán di chuyển

### Bài toán di chuyển

### Bài toán tam giác (IOI 1994)

### Bài toán con kiến

### --Hết--