



BEGIN {Chương trình chính}

Chuẩn bị: đọc dữ liệu và khởi gán một số giá trị;

Tạo bảng;

Tra bảng và in kết quả;

END.

Quy noạch đọng

TaiLieu.vn

PHƯƠNG PHÁP QUY HOACH ĐÔNG: 3.

- 1) Tính nghiệm tối ưu của bài toán trong trường hợp riêng đơn giản nhất.
- Tìm các công thức đệ quy biểu diễn nghiệm tối ưu của bài toán lớn thông qua nghiệm tối ưu của các bài toán con.
- 3) Tính nghiệm tối ưu từ dưới lên (bottom up) và ghi lại các nghiệm tối ưu của các bài toán con đã tính để sử dụng sau này.

VÍ DŲ: 4.

4.1. Dãy Fibonaci:

Đề bài: In ra màn hình 20 số hạng đầu của dãy Fibonaci.

Biết: $F_1 = 1$

 $F_2 = 1$

 $F_3 = 2$

 $F_4 = 3$

 $F_i = F_{i-1} + F_{i-2} \text{ v\'et } i > 2$

I 4 2/15 > Zoom - +

Tran Quang Qua

Mormal 🚐

Giải thuật:



Mười số hạng đầu của dãy Fibonaci:

 $F_i = F_{i-1} + F_{i-2} \ v \acute{\sigma} i \ i > 2$

i	1	2	3	4	5	6	7	8	9	10
F[i]	1	1	2	3	5	8	13	21	34	55

4.2. Tổ hợp chập k của n phần tử:

 \underline{D} ề bài: Tính các phần tử của mảng $C[n, k] = C_n^k = số tổ hợp chập k$ của n phần tử, với $0 \le k \le n \le 20$.

Biết

Tài Liệu Liên Quan

<> Embed

$$C_n^0 = C_n^n = 1$$

$$C_n^0 = C_n^n = 1$$
 $C_n^k = C_{n-1}^{k-1} + C_{n-1}^k$



3

Trần Quang Quá

I (3/15) Zoom - +

Normal

Giải thuật:

1) Tính nghiệm của bài toán trong trường hợp riêng đơn giản nhất.

2) Tìm các công thức đệ quy biểu diễn nghiệm tối ưu của bài toán lớn thông qua nghiệm tối ưu của các bài toán con.

n	k	0	1	2	3	4	5
1		1	1				
2		1	2	1			
3		1	3	3	1		
4		1	4	6	4	1	
5		1	5	10	10	5	1

3) Có thể cải tiến: dùng 2 mảng một chiều thay cho 1 mảng hai chiều.



4.3. Tìm dãy con không giảm dài nhất:

<u>Đề bài</u>: Cho một dãy n số nguyên. Hãy loại bỏ khỏi dãy một số phần tử để được một dãy con không giảm dài nhất. In ra dãy con đó.

Ví du: Input:

10

2 6 -7 5 8 1 -3

-3 5 15

9

Kết quả tìm được dãy con không giảm dài nhất có 4 phần tử:

Giải thuật:

1) <u>Tổ chức dữ liệu:</u>

Gọi A là dãy ban đầu.

Gọi B[i] là số phần tử của dãy con dài nhất trong dãy có i phần tử đầu tiên

A[1] .. A[i] và A[i] được chọn làm phần tử cuối. $(i \in [1, n])$

C là dãy con không giảm dài nhất tìm được.

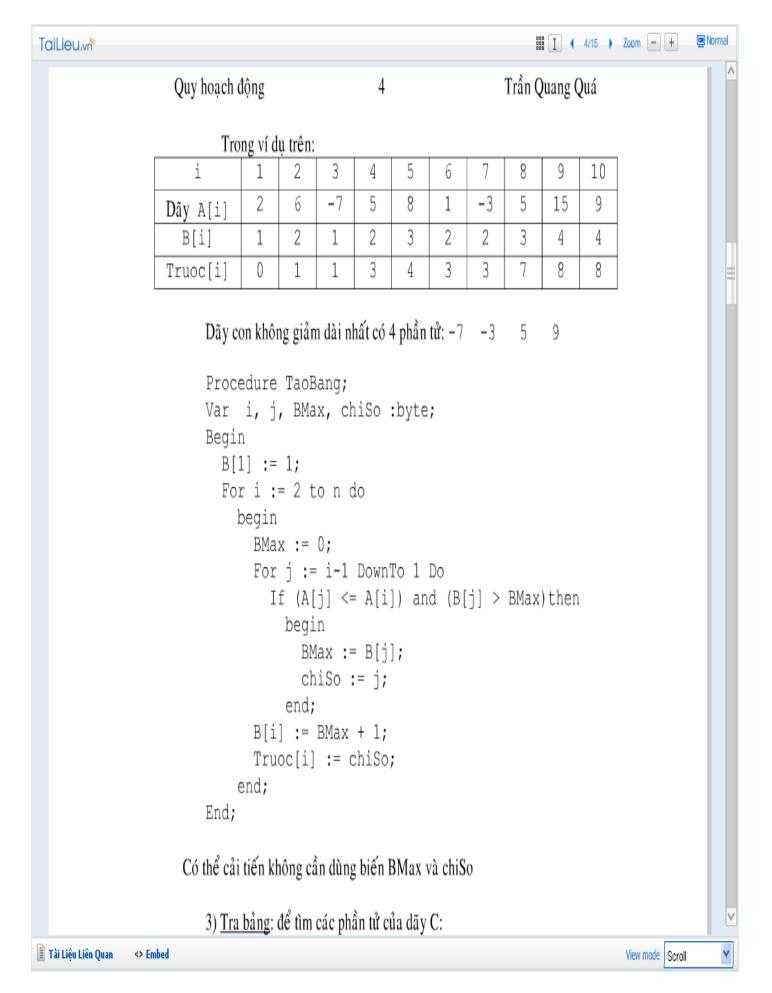
Truoc[i] là chỉ số của phần tử trước phần tử i (các phần tử giữ lại C).

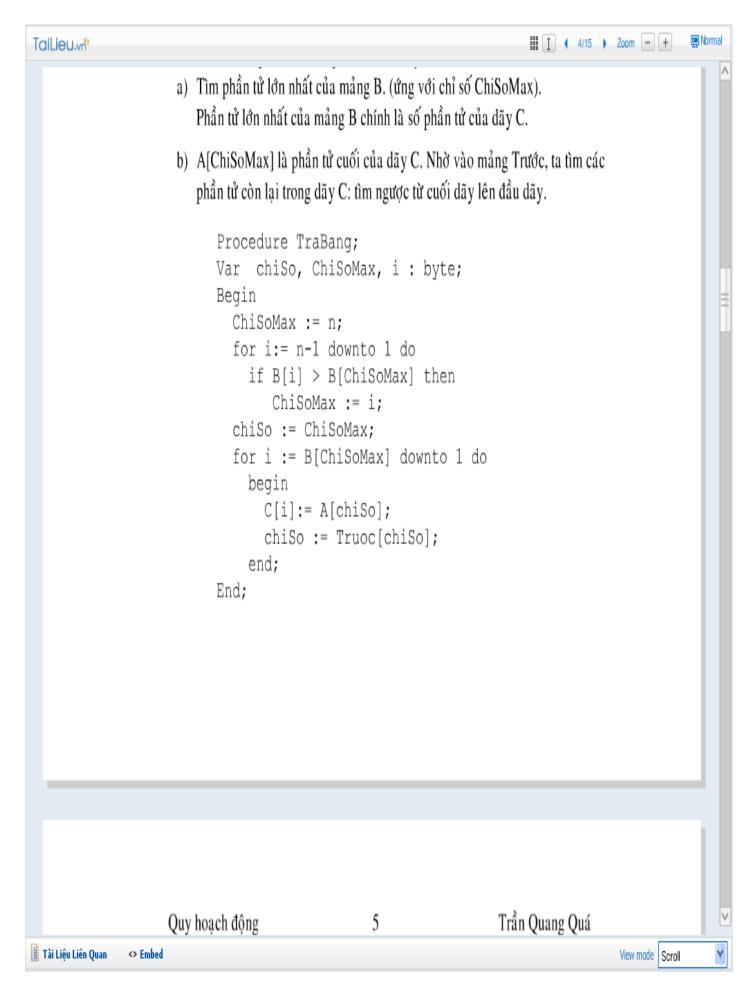
2) Giải thuật tạo bảng: (Tính mảng B và mảng Trước)

Trường hợp đơn giản nhất: dãy chỉ có 1 phần tử, thì B[1] := 1;

For itừ 2 đến n Do

- Xét với mọi j < i và A[j] <= A[i], tìm B[j] lớn nhất (gọi là BMax).
- B[i] := BMax + 1;
- Trước[i] := j; {j là chỉ số ứng với BMax tìm được}.





4.4. Bài toán balô 1:

<u>Dề bài</u>: Cho n món hàng (n ≤ 50). Món thứ i có khối lượng là A[i] (số nguyên). Cần chọn những món hàng nào để bỏ vào một ba lô sao tổng khối lượng của các món hàng đã chọn là lớn nhất nhưng không vượt quá khối lượng W cho trước. (W ≤ 100). Mỗi món chỉ chọn 1 hoặc không chọn.

Input:

Ví du:

OutPut:

Tổng khối lượng của các món hàng bỏ vào ba lô.

Khối lượng của các món hàng đã chọn.

Trong ví dụ trên:

Tổng khối lượng của các món hàng bỏ vào ba lô là 10

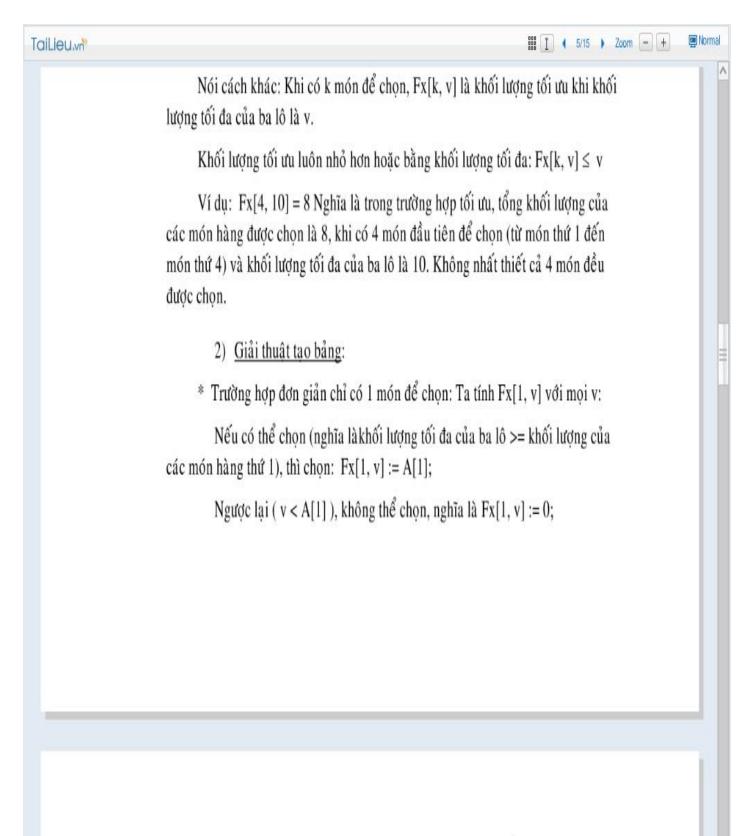
Khối lượng các món hàng được chọn: 5

<u>Hướng giải:</u>

1) Tổ chức dữ liệu:

Fx[k, v] là tổng khối lượng của các món hàng bỏ vào ba lô khi có k món hàng đầu tiên để chọn và khối lượng tối đa của ba lô là v.

 $V \acute{\sigma} i k \in [1, n], v \in [1, W].$



Quy hoạch động

6

Trần Quang Quá

* Giả sử ta đã tính được $Fx[k-1\,,\,v\,]$ đến dòng $k-1,\,$ với mọi $v\in[1,W].$



View mode

Khi có thêm món thứ k để chọn, ta cần tính Fx[k, v] ở dòng k, với mọi v∈[1,W]

Nếu có thể chọn món hàng thứ k ($v \ge A[k]$), thì có 2 trường hợp:

- Trường hợp 1: Nếu chọn thêm món thứ k bỏ vào ba lô, thì

$$Fx[k, v] := Fx[k-1, u] + A[k];$$

Với u là khối lượng còn lại sau khi chọn món thứ k. u = v - A[k]

- Trường hợp 2: Ngược lại, không chọn món thứ k, thì

$$Fx[k, v] := Fx[k-1, v];$$

Trong 2 trường hợp trên ta chọn trường hợp nào có Fx[k, v] lớn hơn.

Ngược lại (v < A[k]), thì không thể chọn, nghĩa là Fx[k, v] := Fx[k-1, v];

Tóm lại: công thức đệ quy là:

Dưới đây là bảng Fx[k,v] tính được trong ví du trên:

k v	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	5	5	5	5	5	5
2	0	2	2	2	5	5	7	7	7	7
3	0	2	2	4	5	6	7	7	9	9
4	0	2	3	4	5	6	7	8	9	10

Procedure TaoBang;

Var k ,v : integer;

Begin

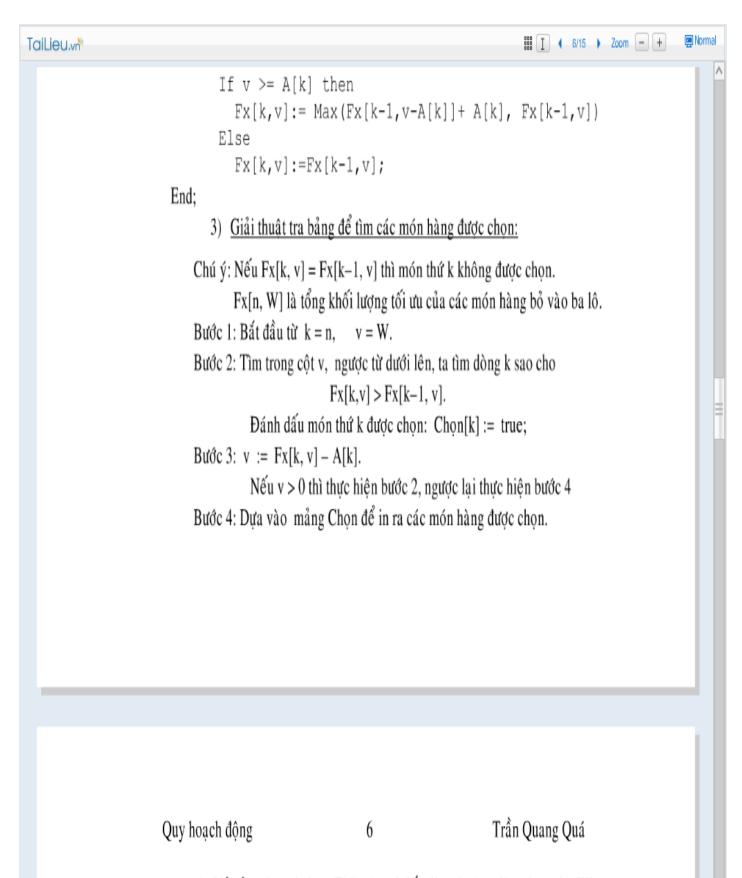
For v:=1 to W do

If $v \ge A[1]$ then Fx[1, v] := A[1]

Else Fx[1, v] := 0;

For k := 2 to n do

for v:=1 to W do



* Giả sử ta đã tính được Fx[k-1, v] đến dòng k-1, với mọi $v \in [1, W]$. Khi có thêm món thứ k để chọn, ta cần tính Fx[k, v] ở dòng k, với mọi $v \in [1, W]$





```
var k, v: Integer;
Begin
  k := n;
  v := w;
  FillChar(chon, SizeOf(chon), false);
  Repeat
    While Fx[k,v] = Fx[k-1,v] do Dec(k);
    chon[k]:= True;
    v := Fx[k,v] - A[k];
  Until v = 0;
  For k := 1 to n do
    If chon[k] then Write(A[k]:5);
  Writeln;
End;
```

4.5. Bài toán chia kẹo:

<u>Đề bài</u>: Cho n gói kẹo (n ≤ 50). Gói thứ i có A[i] viên kẹo. Cần chia các gói kẹo này cho 2 em bé sao cho tổng số viên kẹo mỗi em nhận được chênh lệch ít nhất. Mỗi em nhận nguyên gói. Không mở gói kẹo ra để chia lại.

Hãy liệt kê số kẹo trong các gói kẹo mỗi em nhận được.

Input:

n

A[1] A[2] ... A[n]

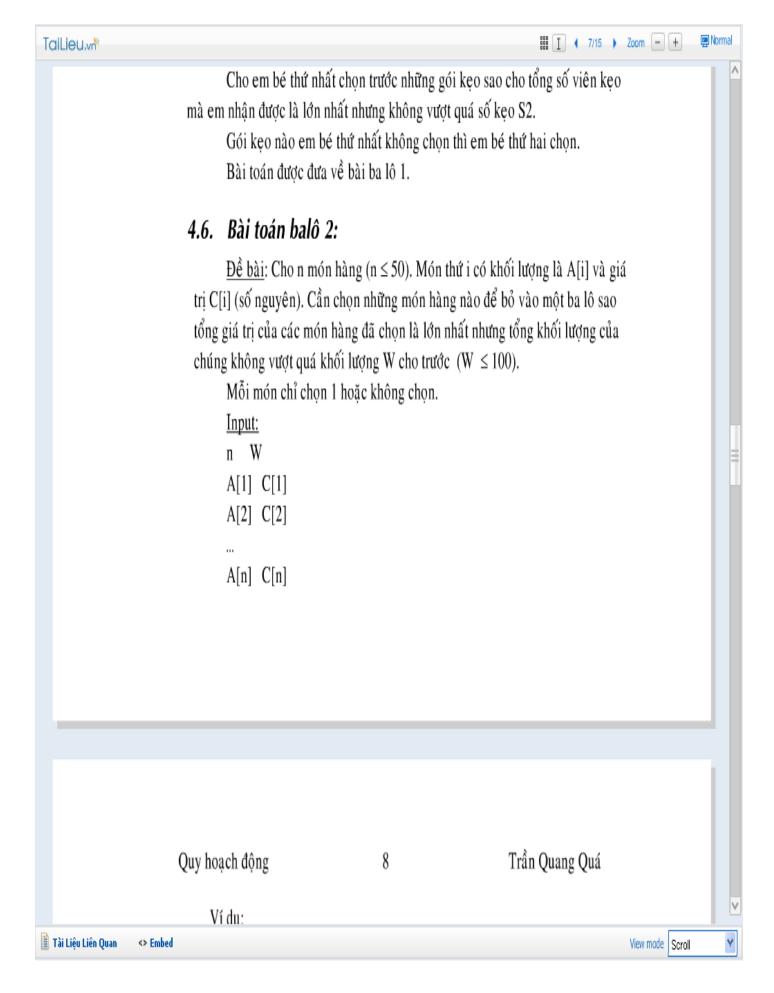
Output: Số kẹo trong các gói kẹo mỗi em nhận được, và tổng số kẹo mỗi em nhận được.

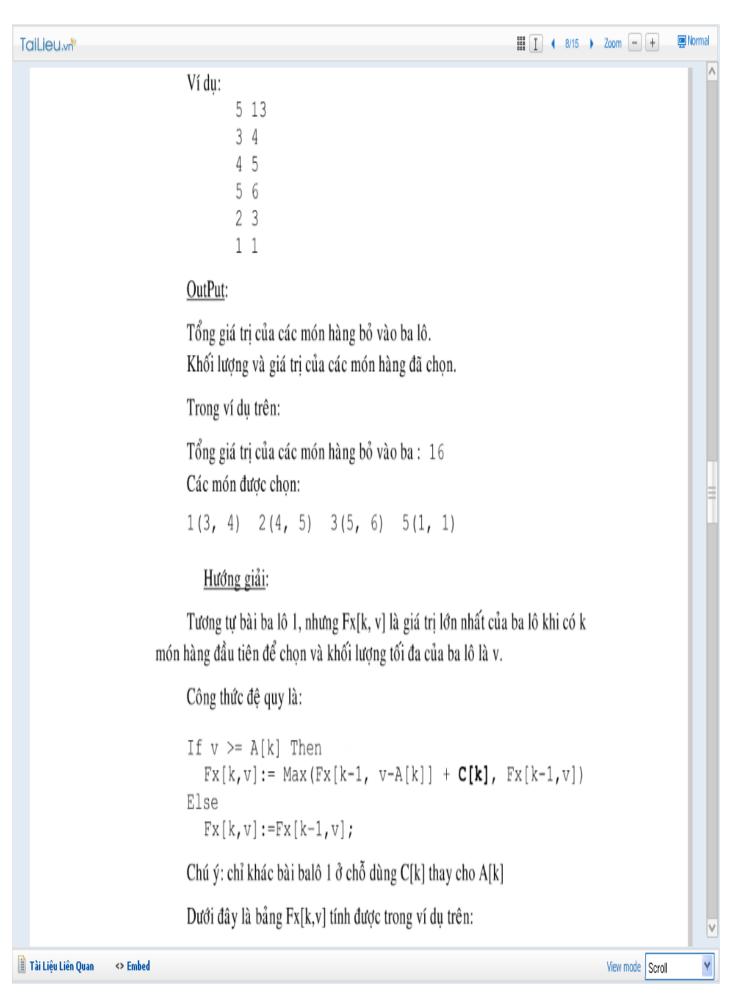
Hướng giải:

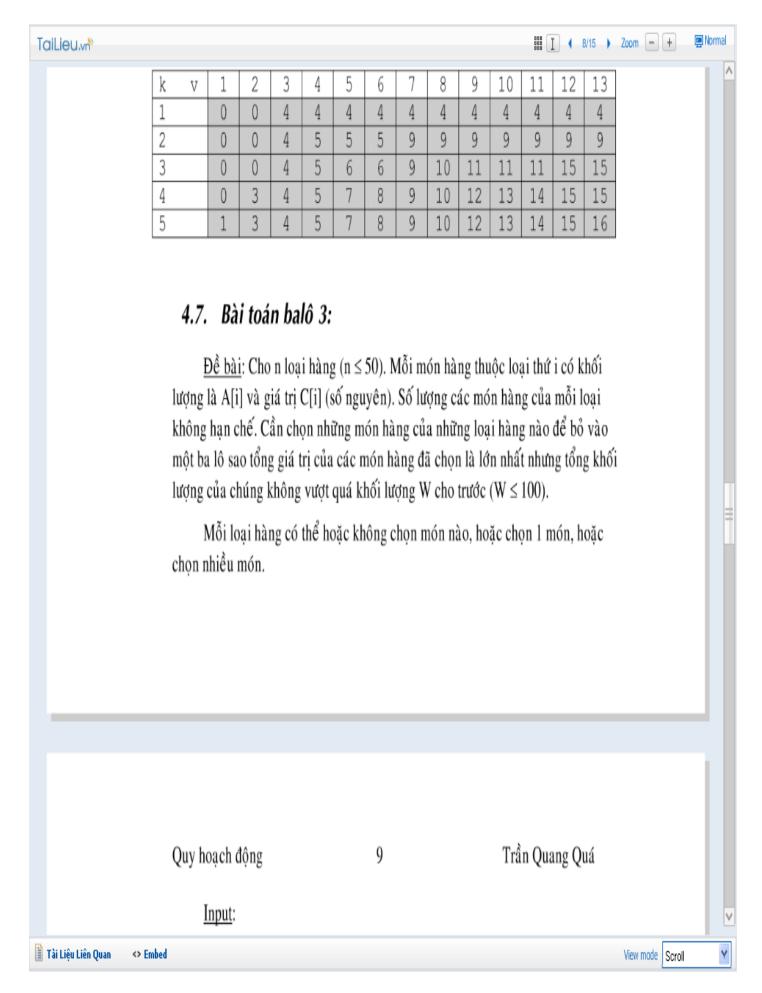
Gọi S là tổng số viện kẹo S := A[1] + A[2] + ... + A[n];

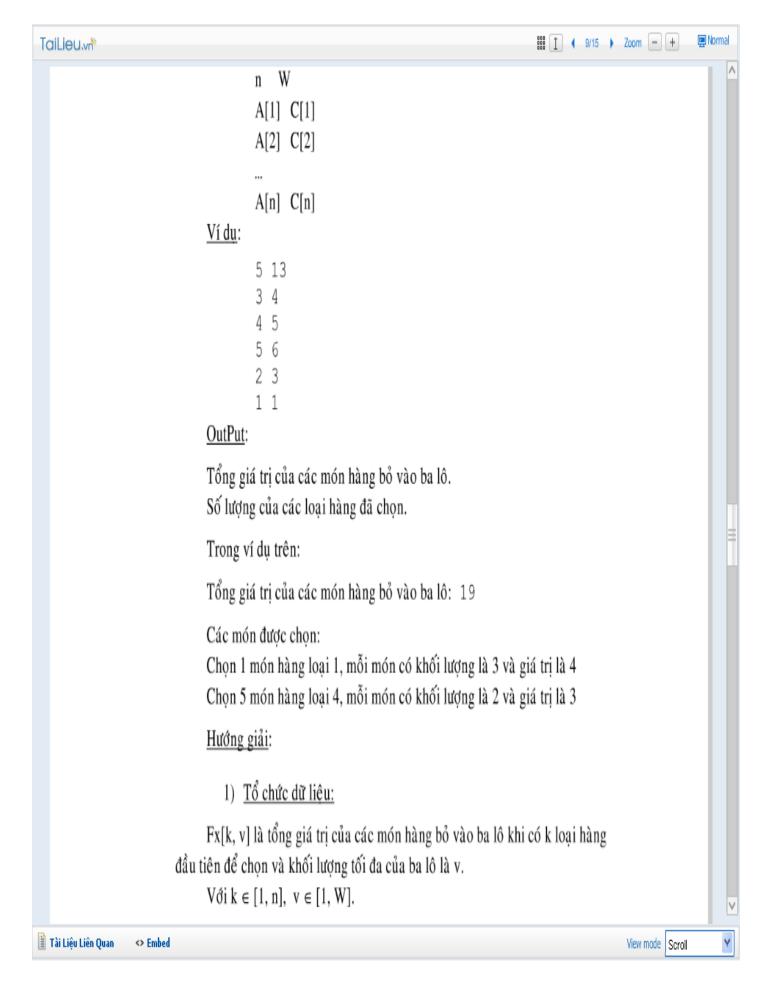
S2 là nửa tổng số kẹo: S2 := S div 2;

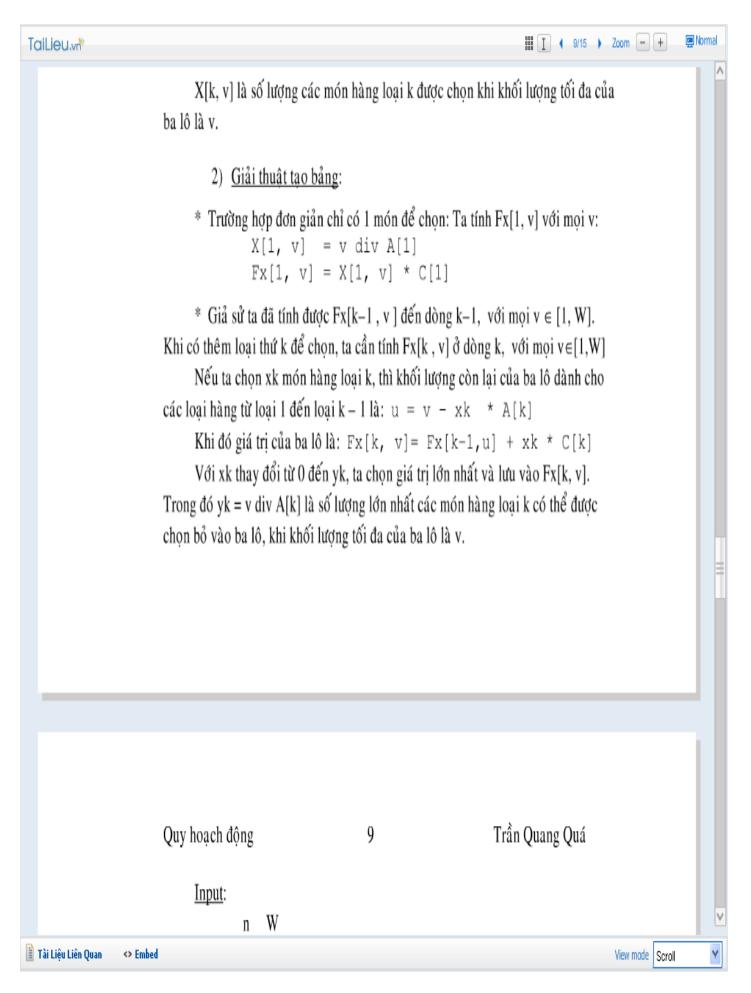
🗎 Tài Liệu Liên Quan











Quy hoạch động

10

Trần Quang Quá

Tóm lại: công thức đệ quy là:

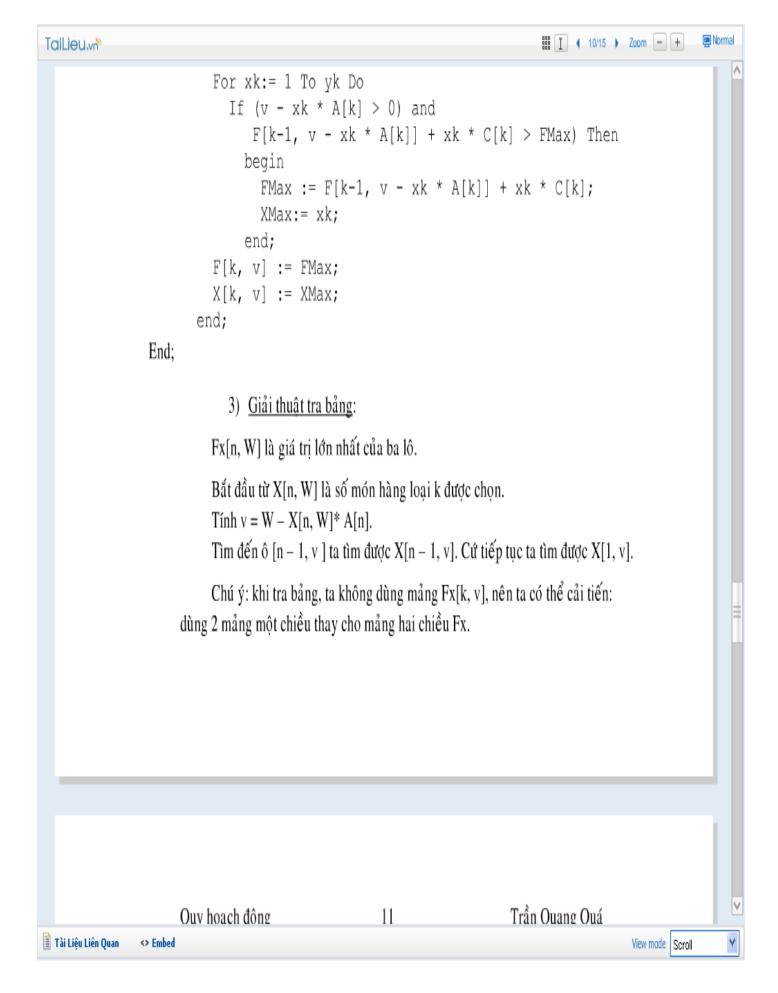
$$Fx[k,v] = Max(Fx[k-1, v - xk * A[k]] + xk * C[k])$$

 $Max x \acute{e}t v \acute{o}i x \acute{k} t hay d \acute{o}i t \acute{u} 0 d \acute{e}n v d i v A[k], v \grave{a} v - xk * A[k] > 0$

Dưới đây là bảng Fx[k,v] và X[k, v] tính được trong ví dụ trên. Bảng màu xám là X[k, v]:

k	V	1		2)	2	3	6	1		j	(,	1		{	}	()	1	()	1	1	1	2	1	3
1		0	0	0	()	4	1	4	1	4	1	8	2	8	2	8	2	12	3	12	3	12	3	16	4	16	4
2		0	0	0	0	4	0	4	0	5	1	8	0	9	1	9	1	12	0	13	1	14	2	16	0	17	1
3		0	0	0	0	4	()	4	0	5	0	8	()	9	0	10	1	12	0	13	0	14	0	16	0	17	0
4		0	0	0	()	4	()	4	()	1	1	8	()	10	2	11	1	13	3	14	2	16	4	17	3	19	5
5		0	0	1	1	4	0	5	1	7	0	8	()	10	0	11	0	13	0	14	0	16	0	17	0	19	0

```
Procedure TaoBang;
Var xk, yk, k: Byte;
   FMax, XMax, v : Word;
Begin
   For v:= 1 To W Do
    begin
    X[1, v] := v div A[1];
```





4.8. Bài toán đổi tiền:

Đề bài: Cho n loại tờ giấy bạc. Tờ giấy bạc thứ i có mệnh giá A[i]. Số tờ mỗi loại không giới hạn. Cần chi trả cho khách hàng số tiền M đồng. Hãy cho biết mỗi loại tiền cần bao nhiều tờ sao cho tổng số tờ là ít nhất. Nếu không đổi được, thì thông báo "KHONG DOI DUOC". N < 50; A[i] < 256; M < 10000

n M Input:

A[1] A[2] ... A[n]

<u>Ví du:</u> 3 18

3 10 12

Tổng số tờ phải trả. Output:

Số tờ mỗi loại.

Cách giải thứ nhất: Tương tự bài ba lô 3

Gọi Fx[i, j] là số tờ ít nhất được dùng để trả số tiền j đồng khi có i loại tiền từ loại 1 đến loại i. Với i = 1 ... n; j = 1 ... M.

X[i, j] là số tờ giấy bạc loại thứ i được dùng chi trả số tiền j đồng.

* Trường hợp đơn giản chỉ có 1 loại tiền để chọn: Ta tính Fx[1, j] với mọi j

$$Fx[1,j] = \left\{ \begin{array}{ll} j & \text{div A[1]} & \text{n\'eu} \ j \ \text{mod A[1]} = 0 \\ \infty & \text{n\'eu} \ j \ \text{mod A[1]} \neq 0 \ (\text{không đổi được}) \end{array} \right.$$

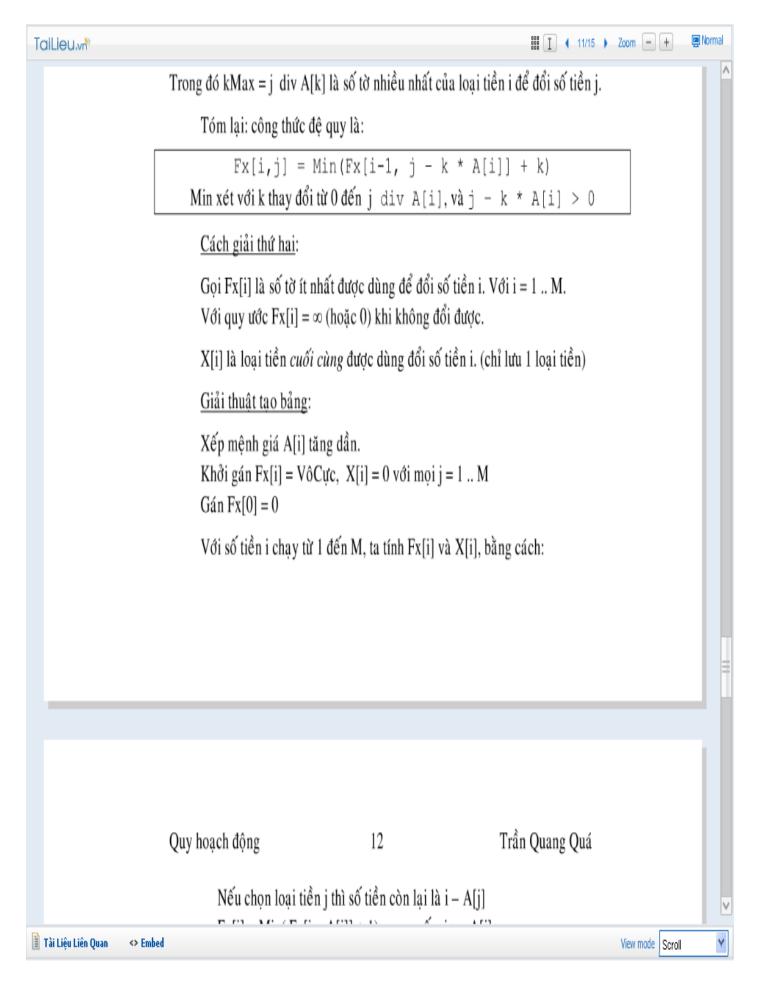
* Giả sử ta đã tính được Fx[i-1, j] đến dòng i-1, với mọi $j \in [1, M]$. Khi có thêm loại tiền thứ i để chọn, ta cần tính Fx[i, j] ở dòng i, với mọi j∈[1, M]

Nếu ta chọn k tờ loại i, thì số tiền còn lại dành cho các loại tiền khác từ loại

$$1 \operatorname{d\acute{e}n} \operatorname{loại} i - 1 \operatorname{là}: u = j - k * A[k]$$

Khi đó tổng số tờ là: Fx[i, j] = Fx[i-1, u] + k

Với k thay đổi từ 0 đến kMax, ta chọn giá trị nhỏ nhất và lưu vào Fx[i, j].



Fx[i] = Min(Fx[i - A[j]] + 1) $n\acute{e}u i >= A[j]$ Min xét với loại tiền j chạy từ 1 đến n. X[i] = j ứng với giá trị min của Fx[i]

Dưới đây là mảng Fx[i] và X[i] tính được trong ví dụ trên (dùng 3 loại tiền 3 đồng, 10 đồng, 12 đồng để đổi số tiền 18 đồng)

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Fx[i]	∞	∞	∞	1	∞	∞	2	∞	∞	3	1	∞	1	2	∞	2	3	∞	3
X[i]	0	0	0	1	0	0	1	0	0	1	2	0	3	2	0	3	2	0	3

Procedure TaoBang;

```
Var i: Word;
    j: Byte;
Begin
  For i := 1 to M Do Fx[i] := VoCuc; \{V\hat{o}Cuc = MaxInt - 1\}
  Fx[0] := 0;
  FillChar(X, SizeOf(X), 0);
  For i := 1 to M Do
                                     { i là số tiền }
                                     {j là loại tiền }
    For j := n DownTo 1 Do
        If i >= A[j] Then
           If (Fx[i] > Fx[i - A[j]] + 1) Then
             Begin
               Fx[i] := Fx[i - A[j]] + 1;
               X[i] := j;
             End;
End;
```

📋 Tài Liêu Liên Quan

<> Embed



4.9. Phân công kĩ sử

(Đề thi tuyển sinh sau Đại học khoá 1997 Đại học Tổng hợp Tp HCM)

13

Một cơ sở phần mềm có n phòng máy vi tính. Cơ sở này phải tuyển chọn m kĩ sư để bảo trì máy. Sau khi tham gia ý kiến của các chuyên gia và kinh nghiệm của các đơn vị khác, người ta hiểu rằng nếu phân công i kĩ sư chuyên bảo trì tại phòng máy j thì số máy hỏng hằng năm phải thanh lí là a[i,j]. Do hạn chế về thời gian và điều kiện đi lại chỉ có thể phân công mỗi kĩ sư bảo trì tại một phòng máy. Bảng ví dụ dưới đây với m = 5 (kĩ sư) và n = 3 (phòng máy).

Số kĩ sư	phòng máy 1	phòng máy 2	phòng máy 3
0	14	25	20
1	10	19	14
2	7	16	11
3	4	14	8
4	1	12	6
5	0	11	5

Yêu cầu: Tìm ra phương án phân công mỗi phòng máy phân bao nhiêu kĩ sư sao cho tổng số máy phải thanh lí hằng năm là ít nhất.

Dữ liệu: vào từ file văn bản KiSu.inp có 2 dòng:

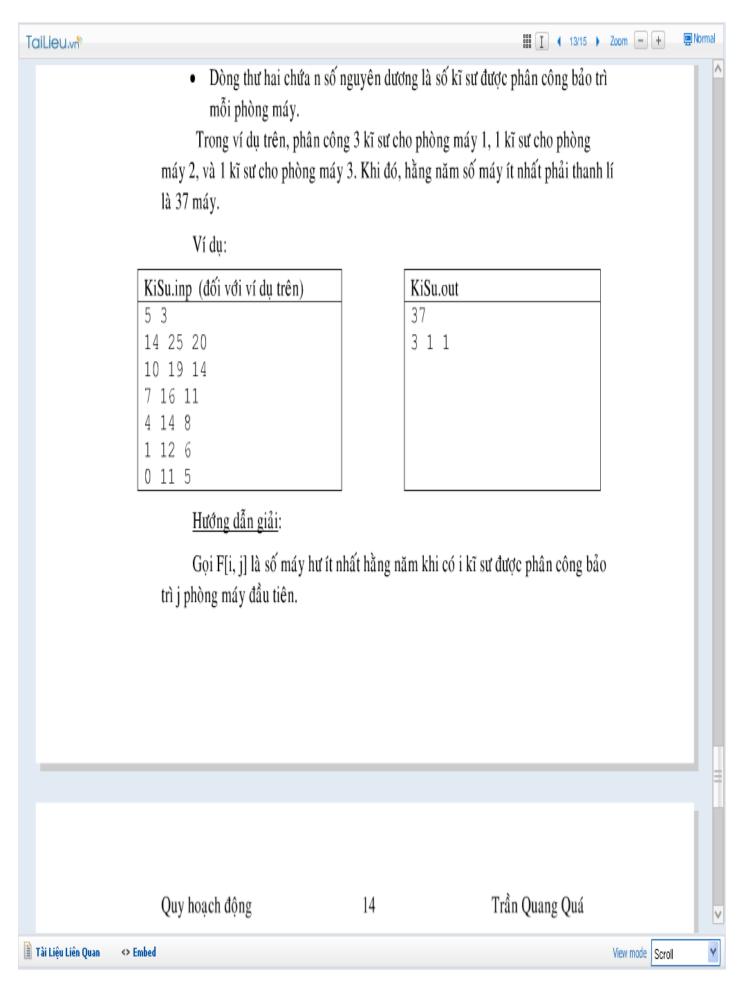
- dòng đầu gồm 2 số nguyên dương m, n. (m, n < 50)
- m+1 dòng tiếp theo bảng a[i, j].

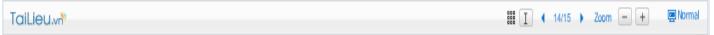
Kết quả: đưa ra file văn bản KiSu.out gồm 2 dòng:

• Dòng đầu chứa tổng số máy (ít nhất) phải thanh lí hằng năm.









4.10. Tam phân đa giác:

Cho một đa giác lồi n đỉnh. Hãy phân đa giác này thành n – 2 tam giác bằng n – 3 đường chéo, sao cho tổng của độ dài của các đường chéo này là nhỏ nhất. Các đường chéo này không cắt nhau (chỉ có thể giao nhau ở đỉnh của đa giác).

Dữ liệu: vào từ file văn bản TAMPHAN.INP có n + 1 dòng:

- Dòng đầu chứa một số nguyên n là số đỉnh của đa giác (3 < n < 50).
- Mỗi dòng trong n dòng kế tiếp chứa hai số thực là hoành độ và tung độ của mỗi đính của đa giác.

Kết quả: đưa ra file văn bản TAMPHAN.OUT, gồm dòng đầu chứa một số thực (có 4 chữ số thập phân) là tổng nhỏ nhất của độ dài của các đường chéo. Mỗi dòng trong n – 3 dòng tiếp theo chứa 2 số nguyên là chỉ số của hai đỉnh của mỗi đường chéo được chọn.

Ví dụ:

TAMPHAN.INP
6
2 1
2 4
6 6
10 6
10 3
7 0

TAMPHAN.OUT							
17.4859							
2 6							
3 6							
3 5							

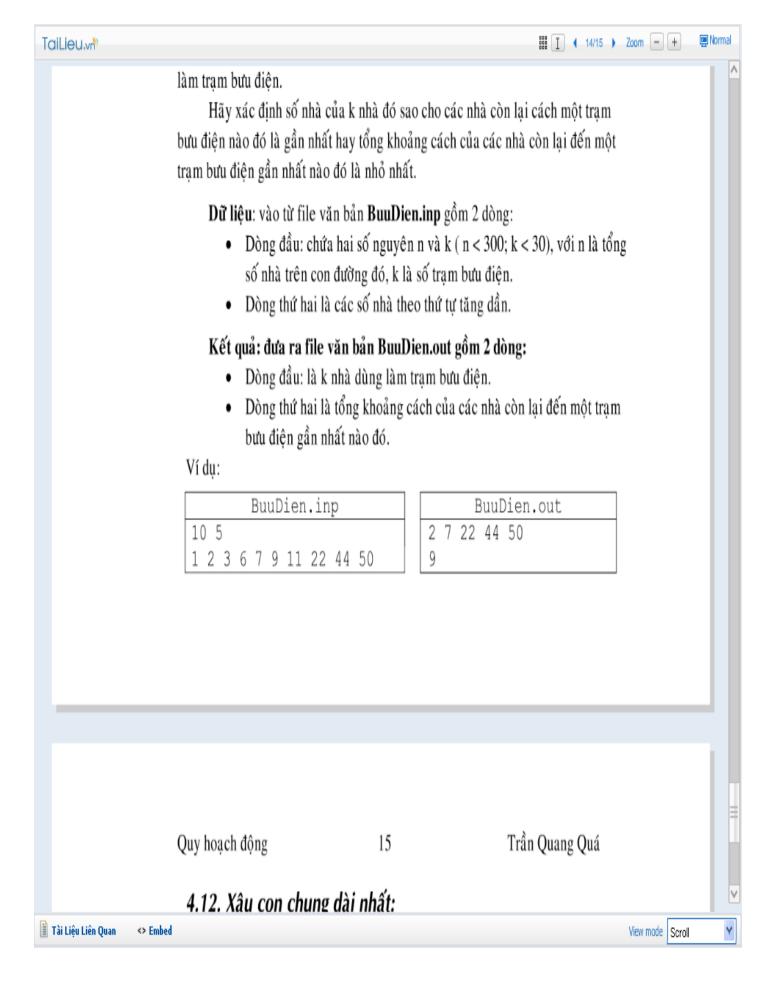
<u>Hướng dẫn</u>: Gọi Fx[i, j] là tổng độ dài ngắn nhất của các đường chéo khi tam phân đa giác có i đỉnh kể từ đỉnh thứ j.

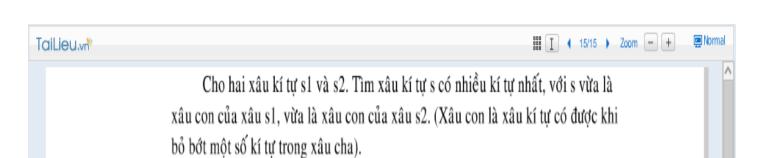
4.11. Trạm bưu điện

Trên một con đường thẳng, dài, số nhà của những nhà dọc theo một bên đường là số đo độ dài tính từ đầu con đường (số nguyên). Người ta chọn ra k nhà









 ${f D}{f w}$ liệu: vào từ tập tin văn bản ${f XauChung.inp}$ gồm hai dòng, mỗi dòng là một xâu kí tự.

Kết quả: đưa ra tập tin văn bản XauChung. out gồm 2 dòng:

- Dòng đầu là độ dài của xâu con chung dài nhất.
- Dòng thứ hai là xâu con chung s.

Ví dụ:

	Х	auChu	ng.in	p	
luong	the	vinh	bien	hoa	
ngo qı	ıyen	dong	nai		

		Xá	auChung.out
9			
ng	en	n	a

