**Bài toán: Biến đổi xâu**

Cho xâu kí tự X, xét 3 phép biến đổi:

1. Phép chèn 1 kí tự vào sau vị trí i của xâu X
2. Phép xóa kí tự tại vị trí i của xâu X
3. Phép thay thế kí tự tại vị trí I của xâu X

Yêu cầu: Cho trước xâu Y, hãy tìm một số ít nhất các phép biến đổi trên để biến xâu X thành xâu Y

Input:

X, Y

Output:

Số phép biến đổi

Các phép biến đổi

**Thuật giải:**

1. Công thức truy hồi

Giả sử m là độ dài xâu X và n là độ dài xâu Y. Gọi C[i, j] là số phép biến đổi tối thiểu để biến xâu gồm i ký tự đầu của xâu X: X1X2 … Xi thành xâu gồm j ký tự đầu của xâu Y: Y1Y2…Yj.

Ta nhận thấy rằng X = X1X2…Xm và Y = Y1Y2…Yn nên:

* Nếu Xm = Yn thì ta chỉ cần biến đoạn X1X2…Xm-1 thành Y1Y2…Yn-1 tức là trong trường hợp này:

**C[m, n] = C[m - 1, n - 1].**

* Nếu Xm ≠ Yn thì tại vị trí Xm ta có thể sử dụng một trong 3 phép biến đổi:
* ***Hoặc chèn vào sau vị trí m của X, một ký tự đúng bằng Yn:***

Thì khi đó C[m, n] sẽ bằng 1 phép chèn vừa rồi cộng với số phép biến đổi biến dãy X1…Xm thành dãy Y1…Yn-1: C[m, n] = 1 + C[m, n - 1]

* ***Hoặc thay vị trí m của X bằng một ký tự đúng bằng Yn***

Thì khi đó C[m, n] sẽ bằng 1 phép thay vừa rồi cộng với số phép biến đổi biến dãy X1…Xm-1 thành dãy Y1…Yn-1: C[m, n] = 1 + C[m-1, n - 1]

* ***Hoặc xoá vị trí thứ m của X***

Thì khi đó C[m, n] sẽ bằng 1 phép xoá vừa rồi cộng với số phép biến đổi biến dãy X1…Xm-1 thành dãy Y1…Yn: C[m, n] = 1 + C[m-1, n]

Vì C[m, n] phải là nhỏ nhất có thể, nên trong trường hợp Xm ≠ Yn thì:

C[m, n] = min(C[m, n - 1], C[m - 1, n - 1], C[m - 1, n]) + 1.

Ta xây dựng xong công thức truy hồi.

1. **Cơ sở quy hoạch động**

• C[0, j] là số phép biến đổi biến xâu rỗng thành xâu gồm j ký tự đầu của C. Nó cần tối thiểu j phép chèn: C[0, j] = j

• C[i, 0] là số phép biến đổi biến xâu gồm i ký tự đầu của S thành xâu rỗng, nó cần tối thiểu i phép xoá: C[i, 0] = i

Vậy đầu tiên bảng phương án C (cỡ[0..m, 0..n]) được khởi tạo hàng 0 và cột 0 là cơ sở quy hoạch động. Từ đó dùng công thức truy hồi tính ra tất cả các phần tử bảng B.

Sau khi tính xong thì C[m, n] cho ta biết số phép biến đổi tối thiểu.

c) Tìm kết quả tối ưu

C[M][N] là số phép biến đổi ít nhất biến X thành Y.

1. **Truy vết:**

• Nếu Xm = Yn thì chỉ việc xét tiếp C[m - 1, n - 1].

• Nếu không, xét 3 trường hợp:

♦ Nếu C[m, n] = C[m, n - 1] + 1 thì phép biến đổi đầu tiên được sử dụng là: chèn kí tự Yn vào sau vị trí m

♦ Nếu C[m, n] = C[m - 1, n - 1] + 1 thì phép biến đổi đầu tiên được sử dụng là: thay thế kí tự tại vị trí m bằng kí tự Yn

♦ Nếu C[m, n] = C[m - 1, n] + 1 thì phép biến đổi đầu tiên được sử dụng là:

xóa kí tự tại vị trí thứ m.

Đưa về bài toán với m, n nhỏ hơn truy vết tiếp cho tới khi về C[0, 0]

1. Thiết kế:

/\*Tạo cơ sở quy hoạch động\*/

for i = 0…M

c[i][0] = i

for j = 0…N

c[0][j] = j

/\*Xây dựng bảng\*/

for i = 0…M

for j=0…N

if(X[i-1] = Y[j-1])

C[i][j] = C[i-1][j-1]

else

c[i][j]=Min(c[i][j-1],c[i-1][j-1],c[i-1][j])+1;

/\*Số phép biến đổi ít nhất\*/

C[M][N]

/\*Truy vết\*/

while (m>0 || n>0)

if (x[m-1] == y[n-1])

m=m-1

n=n-1 else

if (n>0 && c[m][n]-1==c[m][n-1])

Chèn Y[n-1] vào sau vị trí m

n=n-1

else if (m>0 && c[m][n]-1==c[m-1][n])

Xóa ký tự tại vị trí m

m=m-1

else

Thay thế ký tự tại vị trí m bằng Y[n-1]

x[m-1]=y[n-1];

m=m-1

n=n-1

**Bài toán: Đổi tiền**

Có n loại tiền xu: x[1], x[2],...x[N]. Yêu cầu đổi số tiền M sao cho số đồng xu cần dùng là ít nhất.

Thuật toán:

Gọi L[i][j] là số đồng xu ít nhất cần dùng để đổi số tiền j

Vậy L[n][M] là số đồng xu its nhất cần dùng để đổi được số tiền M

Input:

N

x[1]…x[N]

M

Output:

Số tờ tiền phải dùng

Các tờ tiền đã dùng

1. **Công thức truy hồi**

Giả sử ta đang xét đến đồng xu thứ i:

Nếu không chọn đồng thứ i (x[i] > j)

L[i][j] = L[i - 1][j];

Ngược lại là chọn đồng thứ j:

L[i][j] = L[i][j - x[i]] + 1;

Do phải chọn ra số đồng xu ít nhất nên phải so sánh với cách đổi số tiền j từ đồng thứ 1 đến đồng thứ (i-1) nên ta phải lấy số đồng ít nhất, có nghĩa là:

L[i][j] = Min(L[i-1][j], L[i][j - x[i] ] + 1)

Vậy công thức truy hồi là:

Nếu không chọn đồng i ( j < x[i] )

L[i][j] = L[i-1][j];

Ngược lại:

L[i][j] = Min(L[i-1][j], L[i][j - x[i] ] + 1)

1. **Cơ sở quy hoạch động**

Ban đầu chưa có đồng xu nào mà số tiền cần đổi là j nên số đồng xu ít nhất là vô cùng:

L[0][j] = vô cùng , với j=1,..M

Ban đầu cũng phải chọn ra các đồng xu sao cho đổi được số tiền là 0 nên số đồng cần chọn ít nhất là 0 đồng

L[i][0] = 0

1. **Tìm kết quả tối ưu**

L[N, M] = số tờ tiền ít nhất để đổi M đơn vị tiền được dùng các tờ tiền mệnh giá x1…xN

1. **Truy vết:**

Để tìm các đồng xu được chọn và số đồng sử dụng của mỗi loại.

Nếu L[n][M] = vô cùng – tức là không có phương án để đổi số tiền M.

Ngược lại: Xuất phát từ ô (n,M)

i= n,j = W

Nếu L[i][j] = L[i-1][j] thì truy vết đến ô (i-1,j)

Ngược lại L[i][j] = L[i][j-x[i]] thì truy vết đến ô (i,j-x[i])

Quá trình truy vết dừng khi đến dòng 0 (i=0) hoặc cột 0 (j=0)

1. **Thiết kế:**

/\*cơ sở quy hoạch động\*/

for i=0…m

L[0][i] =vocung

for i=0…n

L[i][0] = 0

/\*Xây dựng bảng\*/

for i = 1…n

for j = 1…m

L[i][j] = L[i-1][j]

if(x[i]<j)

if(L[i][j-x[i]] + 1 < L[i][j])

L[i][j] = L[i][j-x[i]] + 1

/\*Số tờ tiền ít nhất\*/

if(L[n][m] < INFINITY)

print L[n][m]

else

print “Không có cách đổi”

/\*Truy vết\*/

i = n, j = m

while(j > 0)

if(L[i][j] = L[i-1][j])

i = i-1

else

print x[i]

j = j – x[i]