

BUỔI 6: QUY HOẠCH ĐỘNG

Trình độ nhập môn

5.1. Đổi tất cả các số thập phân từ 1 đến n sang hệ nhị phân.

5.2. Tính $C_n^k = \frac{n!}{k!(n-k)!}$ với $0 \leq k \leq n \leq 20$.

5.3. Viết chương trình xuất ra n phần tử đầu tiên của dãy Fibonacci

5.4. Viết hàm xóa các phần tử trùng nhau trong dãy, chỉ giữ lại một phần tử trong đó. Ví dụ: 1 2 3 2 1 2 4 -> 1 2 3 4

5.5. Viết chương trình tính tích của 2 ma trận

5.6. Viết chương trình đếm và liệt kê các mảng con tăng dần trong mảng một chiều các số nguyên.

Ví dụ: 6 5 3 2 3 4 2 7 các dãy con tăng dần là 2 3 4 và 2 7

5.7. Viết chương trình tìm mảng con tăng dần có tổng lớn nhất trong mảng một chiều.

5.8. Viết chương trình in ra tam giác Pascal

5.9. Viết chương trình đếm số lần xuất hiện của từng loại ký tự trong một chuỗi.

5.10. Viết chương trình đảo ngược các từ trong một chuỗi, mỗi từ được định nghĩa là cách nhau ít nhất một ký tự trắng. Ví dụ:

“TU DO HANH PHUC” -> “UT OD HNAH CUHP”

Kỹ thuật lập trình

5.11. (Bài toán tìm dãy con đơn điệu dài nhất). Cho N số nguyên dương, dãy a_1, a_2, \dots, a_n gồm các số thực. Tìm trong dãy đó có một dãy con $a_{i_1}, a_{i_2}, \dots, a_{i_k}$ của dãy trên thỏa mãn điều kiện:

i) $a_{i_j} \leq a_{i_{j+1}}$

ii) $i_j \leq i_{j+1}$

iii) Không có dãy con nào thỏa tính chất i) & ii), mà có nhiều phần tử hơn.

Hướng dẫn:

Đặt $A_0 = -\infty$ và $A_{n+1} = \infty$.

Gọi $L[i]$ là độ dài dãy con tăng dài nhất của các phần tử lấy trong miền từ A_i đến A_{n+1} và phần tử đầu của dãy tăng là A_i . Ta có công thức QHĐ để tính $L[i]$ như sau:

$$+ L[n+1] = 1$$

$$+ L[i] = \max(L[j]) + 1 \text{ với mọi } i < j < n+1 \text{ và } A_i \leq A_j$$

5.12. (Xâu con chung dài nhất) Cho hai xâu văn bản X và Y số độ dài nằm trong khoảng từ 50 đến 100. Xâu Z được gọi là “xâu con chung” của X và Y nếu Z nhận được từ X (hoặc Y) bằng cách loại bỏ một số kí tự nào đó. Ví dụ nếu $X = \text{”AdksHKoiGAdksHKoiG”}$ và $Y = \text{”ADKSHKOIGADKSHKOIG”}$ thì $Z = \text{”AHK”}$ là một con xâu chung của X và Y. Hãy tìm xâu con chung lớn nhất có thể được (trong ví dụ trên Z cực đại sẽ là “AHKGAHKG”)

Hướng dẫn:

Gọi $L[i][j]$ là độ dài xâu con chung dài nhất của xâu $X[i]$ gồm i kí tự phần đầu của X và xâu $Y[j]$ gồm j kí tự phần đầu của Y.

Ta có công thức QHĐ như sau:

$$+ L[0][j] = L[i][0] = 0$$

$$+ L[i][j] = L[i-1][j-1] + 1 \text{ nếu } X[i] = Y[j]$$

$$+ L[i][j] = \max(L[i-1][j], L[i][j-1]) \text{ nếu } X[i] \neq Y[j]$$

5.13. Tính các phần tử của mảng $C[n][k] = C_n^k =$ số tổ hợp chập k của n phần tử, với $0 \leq k \leq n \leq 20$.

$$\text{Biết} \quad C_n^0 = C_n^n = 1 \quad \text{và} \quad C_n^k = C_{n-1}^{k-1} + C_{n-1}^k$$

Hướng dẫn:

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

```
for (i=1;i<=n;i++)  
{  
    C[0][i] = 1;  
    C[i][i] = 1;  
}
```

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.

```
for (i=2;i<=n,i++)  
    for (j=1;j<i,j++)  
        C[i][j] = C[i-1][j-1] + C[i-1][j];
```

C[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

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

5.14. Cho n món hàng ($n \leq 50$). 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. ($W \leq 100$). Mỗi món chỉ chọn 1 hoặc không chọn.

Input:

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

Ví dụ:

4 10
5 2 4 3

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 2 3

Hướng dẫn:

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 .

Với $k \in [1, n]$, $v \in [1, W]$.

Nói cách khác: Khi có k món để chọn, $Fx[k][v]$ là khối lượng tối ưu khi khối lượng tối đa của ba lô là v .

Khối lượng tối ưu luôn nhỏ hơn hoặc bằng khối lượng tối đa: $Fx[k][v] \leq v$

Ví dụ: $Fx[4][10] = 8$, nghĩa là trong trường hợp tối ưu, tổng khối lượng của các món hàng được chọn là 8, khi có 4 món đầu tiên để chọn (từ món thứ 1 đến món thứ 4) và khối lượng tối đa của ba lô là 10. Không nhất thiết cả 4 món đều được chọn.

Giải thuật tạo bảng:

* Trường hợp đơn giản chỉ có 1 món để chọn: Ta tính $Fx[1][v]$ với mọi v :

Nếu có thể chọn (nghĩa là khối lượng tối đa của ba lô \geq khối lượng của các món hàng thứ 1), thì chọn: $Fx[1][v] = A[1]$;

Ngược lại ($v < A[1]$), không thể chọn, nghĩa là $Fx[1][v] = 0$;

* 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 $v \in [1, W]$

Nếu có thể chọn món hàng thứ k ($v \geq A[k]$), 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:

if ($v \geq A[k]$)

$Fx[k][v] = \text{Max}(Fx[k-1][v - A[k]] + A[k], Fx[k-1][v])$

else

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

Dưới đây là bảng $Fx[k][v]$ tính được trong ví dụ 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

Giải thuật tra bảng để tìm các món hàng được chọn:

Chú ý: Nếu $Fx[k][v] = Fx[k-1][v]$ thì món thứ k không được chọn.

$Fx[n][W]$ là tổng khối lượng tối ưu của các món hàng bỏ vào ba lô.

Bước 1: Bắt đầu từ $k = n$, $v = W$.

Bước 2: Tìm trong cột v , ngược từ dưới lên, ta tìm dòng k sao cho

$$Fx[k][v] > Fx[k-1][v].$$

Đánh dấu món thứ k được chọn: $\text{Chon}[k] = \text{true};$

Bước 3: $v = Fx[k][v] - A[k].$

Nếu $v > 0$ thì thực hiện bước 2, ngược lại thực hiện bước 4

Bước 4: Dựa vào mảng, chọn để in ra các món hàng được chọn.

5.15. (Bài toán chia kẹo) Cho n gói kẹo ($n \leq 50$). 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

$A[1] \ A[2] \ \dots \ A[n]$

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

Hướng dẫn:

Gọi S là tổng số viên kẹo $S = A[1] + A[2] + \dots + A[n]$;

$S2$ là nửa tổng số kẹo: $S2 = S/2$; (chia nguyên)

Cho em bé thứ nhất chọn trước những gói kẹo sao cho tổng số viên kẹo mà em nhận được là lớn nhất nhưng không vượt quá số kẹo $S2$.

Gói kẹo nào em bé thứ nhất không chọn thì em bé thứ hai chọn.

Bài toán được đưa về bài ba lô 1.

5.16. (Bài toán ba lô 2) Cho n món hàng ($n \leq 50$). Món thứ i có khối lượng là $A[i]$ và giá trị $C[i]$ (số nguyên). Cần chọn những món hàng nào để bỏ vào một ba lô sao tổng giá trị của các món hàng đã chọn là lớn nhất nhưng tổng khối lượng của chúng không vượt quá khối lượng W cho trước ($W \leq 100$).

Mỗi món chỉ chọn 1 hoặc không chọn.

Input:

n	W
$A[1]$	$C[1]$
$A[2]$	$C[2]$
\dots	
$A[n]$	$C[n]$

Ví dụ:

5	13
3	4
4	5
5	6
2	3
1	1

Output:

Tổng giá trị của các món hàng bỏ vào ba lô.

Khối lượng và giá trị của các món hàng đã chọn.

Trong ví dụ trên:

Tổng giá trị của các món hàng bỏ vào ba : 16

Các món được chọn: 1(3, 4) 2(4, 5) 3(5, 6) 5(1, 1)

Hướng dẫn:

Tương tự bài ba lô 1, nhưng $Fx[k][v]$ là giá trị lớn nhất của ba lô khi có k món hàng đầu tiên để chọn và khối lượng tối đa của ba lô là v .

Công thức đệ quy là:

if ($v \geq A[k]$)

$$Fx[k][v] = \text{Max}(Fx[k-1][v-A[k]] + C[k], Fx[k-1][v])$$

else

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

Chú ý: chỉ khác bài balô 1 ở chỗ dùng $C[k]$ thay cho $A[k]$

Dưới đây là bảng $Fx[k][v]$ tính được trong ví dụ trên:

$[k][v]$	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	0	4	4	4	4	4	4	4	4	4	4	4
2	0	0	4	5	5	5	9	9	9	9	9	9	9
3	0	0	4	5	6	6	9	10	11	11	11	15	15
4	0	3	4	5	7	8	9	10	12	13	14	15	15
5	1	3	4	5	7	8	9	10	12	13	14	15	16

5.17. (Bài toán balô 3) Cho n loại hàng ($n \leq 50$). Mỗi món hàng thuộc loại thứ i có khối lượng là $A[i]$ và giá trị $C[i]$ (số nguyên). Số lượng các món hàng của mỗi loại không hạn chế. Cần chọn những món hàng của những loại hàng nào để bỏ vào một ba lô sao tổng giá trị của các món hàng đã chọn là lớn nhất nhưng tổng khối lượng của chúng không vượt quá khối lượng W cho trước ($W \leq 100$). Mỗi loại hàng có thể hoặc không chọn món nào, hoặc chọn 1 món, hoặc chọn nhiều món.

Input:

n W
 $A[1]$ $C[1]$
 $A[2]$ $C[2]$
...
 $A[n]$ $C[n]$

Ví dụ:

5 13
3 4
4 5
5 6
2 3
1 1

OutPut:

Tổng giá trị của các món hàng bỏ vào ba lô.

Số lượng của các loại hàng đã chọn.

Trong ví dụ trên:

Tổng giá trị của các món hàng bỏ vào ba lô: 19

Các món được chọn:

Chọn 1 món hàng loại 1, mỗi món có khối lượng là 3 và giá trị là 4

Chọn 5 món hàng loại 4, mỗi món có khối lượng là 2 và giá trị là 3

Hướng dẫn:

Tổ chức dữ liệu:

$Fx[k][v]$ là tổng giá trị của các món hàng bỏ vào ba lô khi có k loại hàng đầu tiên để chọn và khối lượng tối đa của ba lô là v , với $k \in [1, n]$, $v \in [1, W]$.

$X[k][v]$ là số lượng các món hàng loại k được chọn khi khối lượng tối đa của ba lô là v .

Giải thuật tạo bảng:

* Trường hợp đơn giản chỉ có 1 món để chọn: Ta tính $Fx[1][v]$ với mọi v :

$$X[1][v] = v/A[1]; \text{ (chia nguyên)}$$

$$Fx[1][v] = X[1][v] * C[1]$$

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

Nếu ta chọn x_k món hàng loại k , thì khối lượng còn lại của ba lô dành cho các loại hàng từ loại 1 đến loại $k-1$ là: $u = v - x_k * A[k]$

Khi đó giá trị của ba lô là: $Fx[k][v] = Fx[k-1][u] + x_k * C[k]$

Với x_k thay đổi từ 0 đến y_k , ta chọn giá trị lớn nhất và lưu vào $Fx[k][v]$.

Trong đó $y_k = v/A[k]$ (chia nguyên) là số lượng lớn nhất các món hàng loại k có thể được chọn bỏ vào ba lô, khi khối lượng tối đa của ba lô là v .

Tóm lại: công thức đệ quy là:

$$Fx[k][v] = \text{Max}(Fx[k-1][v - x_k * A[k]] + x_k * C[k])$$

Max xét với x_k thay đổi từ 0 đến $v/A[k]$ (chia nguyên), và $v - x_k * A[k] > 0$

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	3	4	5	6	7	8	9	10	11	12	13
----------	---	---	---	---	---	---	---	---	---	----	----	----	----

1	0	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	0	4	0	5	0	8	0	9	0	10	1	12	0	13	0	14	0	16	0	17	0
4	0	0	0	0	4	0	4	0	7	1	8	0	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	0	10	0	11	0	13	0	14	0	16	0	17	0	19	0

5.18. (Bài toán đổi tiền): 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 nhiêu tờ sao cho tổng số tờ là ít nhất. Nếu không đổi được, thì thông báo “KHONG DOI DUOC”. $N < 50$; $A[i] < 256$; $M < 10000$

Input: n M
 $A[1]$ $A[2]$... $A[n]$

Ví dụ: 3 18
 3 10 12

Output: Tổng số tờ phải trả.
 Số tờ mỗi loại.

Hướng dẫn: (Cách 1, 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

$$Fx[1][j] = \begin{cases} j \text{ div } A[1] & \text{nếu } j \bmod A[1] = 0 \\ \infty & \text{nếu } j \bmod A[1] \neq 0 \text{ (không đổi được)} \end{cases}$$

* 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 $j \in [1, M]$

Nếu ta chọn k tờ loại i , thì số tiền còn lại dành cho các loại tiền khác từ loại 1 đến loại $i-1$ là: $u = j - k * A[i]$

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 $Fx[i][j]$.

Trong đó $kMax = j \text{ div } A[i]$ là số tờ nhiều nhất của loại tiền i để đổi số tiền j .

Tóm lại: công thức đệ quy là:

$Fx[i,j] = \text{Min}(Fx[i-1, j - k * A[i]] + k)$ <p>Min xét với k thay đổi từ 0 đến $j \text{ div } A[i]$, và $j - k * A[i] > 0$</p>

Hướng dẫn: (Cách 2)

Gọi $Fx[i]$ là số tờ ít nhất được dùng để đổi số tiền i . Với $i = 1 .. M$.

Với quy ước $Fx[i] = \infty$ (hoặc 0) khi không đổi được.

$X[i]$ là loại tiền cuối cùng được dùng đổi số tiền i . (chỉ lưu 1 loại tiền)

Giải thuật tạo bảng:

Xếp mệnh giá $A[i]$ tăng dần.

Khởi gán $Fx[i] = \infty$, $X[i] = 0$ với mọi $j = 1 .. M$

Gán $Fx[0] = 0$

Với số tiền i chạy từ 1 đến M , ta tính $Fx[i]$ và $X[i]$, bằng cách:

Nếu chọn loại tiền j thì số tiền còn lại là $i - A[j]$

$$Fx[i] = \text{Min}(Fx[i - A[j]] + 1) \quad \text{nếu } i \geq A[j]$$

Min xét với loại tiền j chạy từ 1 đến n .

$$X[i] = j \text{ ứng với giá trị min của } Fx[i]$$

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

5.19. (Phân công kỹ sư - Đề thi tuyển sinh sau Đại học khoá 1997 Đại học Tổng hợp TP HCM) Một cơ sở phần mềm có n phòng máy vi tính. Cơ sở này phải tuyển chọn m kỹ 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 kỹ 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 kỹ sư bảo trì tại một phòng máy. Bảng ví dụ dưới đây với $m = 5$ (kỹ 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.
- ☐ Dòng thứ hai chứa n số nguyên dương là số kĩ sư được phân công bảo trì mỗi phòng máy.

Trong ví dụ trên, phân công 3 kĩ sư cho phòng máy 1, 1 kĩ sư cho phòng máy 2, và 1 kĩ sư cho phòng máy 3. Khi đó, hằng năm số máy ít nhất phải thanh lý là 37 máy.

Ví dụ:

KiSu.inp (đối với ví dụ trên)
5 3
14 25 20
10 19 14
7 16 11
4 14 8
1 12 6
0 11 5

KiSu.out
37
3 1 1

Hướng dẫn:

Gọi $F[i][j]$ là số máy hư ít nhất hằng năm khi có i kỹ sư được phân công bảo trì j phòng máy đầu tiên.

5.20. (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 đỉnh 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	TAMPHAN.OUT
6	17.4859
2 1	2 6
2 4	3 6
6 6	3 5
10 6	
10 3	
7 0	

Hướng dẫn: 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 .

5.21. (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 làm trạm bưu điện.

Hãy xác định số nhà của k nhà đó sao cho các nhà còn lại cách một trạm bưu điện nào đó là gần nhất hay tổng khoảng cách của các nhà còn lại đến một trạm bưu điện gần nhất nào đó là nhỏ nhất.

Dữ liệu: vào từ file văn bản **BuuDien.inp** gồm 2 dòng:

- ☐ Dòng đầu: chứa hai số nguyên n và k ($n < 300$; $k < 30$), với n là tổng số nhà trên con đường đó, k là số trạm bưu điện.
- ☐ Dòng thứ hai là các số nhà theo thứ tự tăng dần.

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

- ☐ Dòng đầu: là k nhà dùng làm trạm bưu điện.
- ☐ Dòng thứ hai là tổng khoảng cách của các nhà còn lại đến một trạm bưu điện gần nhất nào đó.

Ví dụ:

BuuDien.inp	BuuDien.out
10 5	2 7 22 44 50
1 2 3 6 7 9 11 22 44 50	9

5.22. Cho hai dãy số $D1$ và $D2$ mỗi dãy có từ 50 đến 150 số nguyên dương. Dãy số $D3$ được gọi là “dãy con chung” của $D1$ và $D2$ nếu $D3$ nhận được từ $D1$ ($D2$) bằng cách loại bỏ một số ≥ 0 số nào đó. Tìm dãy chung con có số lượng số nhiều nhất.

Hướng dẫn: Một dạng khác của bài toán xâu con chung