BÀI TẬP QUY HOẠCH ĐỘNG

Bài 1: Cho dãy số nguyên dương a_1, a_2, …a_n. Hãy chọn một số số hạng của dãy nhưng không được chọn 3 số hạng liền nhau, sao cho tổng giá trị các số hạng được chọn là lớn nhất.

Ví dụ: *A = (6, 10, 13, 9, 8, 1),* có thể chọn *(6, 13, 9, 1)* cách chọn này có tổng giá trị 29, có thể chọn *(6, 10, 9, 8)* cách chọn này có tổng giá trị là 33 là cách chọn có tổng giá trị lớn nhất.

1. Bài toán con thứ *i* gồm các thành phần tứ 1 đến *i*. Có thể phát biểu lại bài toán như sau: Hãy chọn trong các số hạng từ 1 đến *i*, nhưng không được chọn 3 số hạng liền nhau, sao cho tổng giá trị các số hạng được chọn là lớn nhất. Bài toán con thứ *n* là bài toán ban đầu.

2. Với cách chia đó ta có:

+ Với bài toán con 1, chỉ có 1 món quà nên tổng giá trị thu được là a_1, giá trị tối ưu cho bài toán này là a_1.

+ Với bài toán con 2 cũng vậy, chỉ có 2 món quà nên tổng giá trị thu được là a_1+a_2, giá trị tối ưu cho bài toán này là a_1+a_2.

+ Với bài toán con 3 thì sẽ có các cách chọn thỏa mãn yêu cầu của bài toán: (a_1, a_2), (a_1, a_3), (a_2, a_3) với tổng giá trị lần lượt là: a_1+a_2, a_1+a_3, a_2+a_3. Từ đó giá trị tối ưu cho bài toán này là giá trị lớn nhất trong 3 cách chọn: $max\{a_1+a_2, a_1+a_3, a_2+a_3\}$

+ Với bài toán con 4 ta có các cách chọn: (a_1, a_2), (a_1, a_3), (a_1, a_4), (a_2, a_3), (a_2, a_4), (a_3, a_4), (a_1, a_2, a_4), (a_1, a_3, a_4). Từ đó giá trị tối ưu cho bài toán này là:

$max\{a_1+a_2, a_1+a_3, a_1+a_4, a_2+a_3, a_2+a_4, a_3+a_4, a_1+a_2+a_4, a_1+a_3+a_4\}$

$= max\{max\{a_1+a_2, a_1+a_3, a_2+a_3\}, \mathbf{max\{a_1+a_4, a_2+a_4, a_1+a_2+a_4\}}, \mathbf{max\{a_1+a_3+a_4, a_3+a_4\}}\}$.

$= max\{\underline{\mathbf{max\{a_1+a_2, a_1+a_3, a_2+a_3\}}}, \underline{\mathbf{(a_1+a_2)}}+a_4, \underline{\mathbf{(a_1)}}+a_3+a_4\}$.

BT con 3 **BT con 2** **BT con 1**

(Do a_1, a_2, a_3, a_4 nguyên dương nên: $max\{a_1+a_4, a_2+a_4, a_1+a_2+a_4\} = a_1+a_2+a_4$ và $max\{a_1+a_3+a_4, a_3+a_4\} = a_1+a_3+a_4$)

Có thể thấy lời giải tối ưu của bài toán con thứ tư phụ thuộc lời giải tối ưu của các bài toán con 1, 2 và 3. Hay bài toán có cấu trúc con tối ưu: lời giải tối ưu của 3 bài toán con thứ *i-1*, *i-2* và *i-3* sẽ cho lời giải tối ưu cho bài toán con thứ *i*, lời giải tối ưu của 3 bài toán con thứ *n-1*, *n-2* và *n-3* sẽ cho lời giải tối ưu cho bài toán con thứ *n* – bài toán ban đầu.

Gọi *f(i)* là giá trị tối ưu của bài toán con thứ *i*, là tổng giá trị lớn nhất có thể chọn được trong các món quà từ món quà thứ nhất đến món quà thứ *i*. Ta có:

$f(1)=a_1$.

$f(2)=a_1+a_2$.

$f(3)=max\{a_1+a_2, a_1+a_3, a_2+a_3\}$

$f(4)=max\{max\{a_1+a_2, a_1+a_3, a_2+a_3\}, (a_1+a_2+a_4), (a_1)+a_3+a_4\}$

$=max\{f(3), f(2)+a_4, f(1))+a_3+a_4\}$.

…

$f(i)=max\{f(i-1), f(i-2)+a_i, f(i-3)+a_{i-1}+a_i\}$.

Từ đó ta có công thức:

$$f(i) = \begin{cases} a_1 & khi\ i = 1 \\ a_1 + a_2 & khi\ i = 2 \\ max\{a_1 + a_2, a_1 + a_3, a_2 + a_3\} & khi\ i = 3 \\ max\{f(i-1), f(i-2) + a_i, f(i-3) + a_{i-1} + a_i\} & khi\ i > 3 \end{cases}$$

Gọi mãng *a[1..N]* lưu dãy đầu vào (input). Ta có đoạn chương trình đệ quy tính giá trị tối ưu *f(n)* cho bài toán.

```
function f(i : longint) : longint;
begin
        if i = 1 then f := a[1]
        else if i = 2 then f := a[1] + a[2]
                else if i = 3 then f := max(a[1] + a[2], a[1] + a[3], a[2] + a[3])
                        else f := max(f(i - 1), f(i - 2) + a[i], f(i - 3) + a[i - 1] + a[i]);
end;
BEGIN
…
write(f(n));
…
END.
```

Bài 2: **Số 0 tận cùng**

Cho trước một dãy số nguyên dương gồm n phần tử. Nhiệm vụ của bạn rất đơn giản, hãy chọn ra k phần tử từ dãy số đã cho sao cho tích của chúng có nhiều chữ số 0 tận cùng bên phải nhất có thể.

Dữ liệu : Vào từ file văn bản LASTZERO.INP gồm:

-Dòng thứ nhất gồm 2 số nguyên dương n, k ($k \le n$).

-Dòng thứ hai gồm n số nguyên dương $a_1, a_2, \ldots, a_n$ ($a_i \le 10^{18}$) là dãy số cho trước.

Kết quả: Đưa ra file văn bản LASTZERO.OUT gồm một dòng ghi một số nguyên dương là số chữ số 0 tận cùng bên phải của tích k phần tử được chọn trong cách chọn tối ưu nhất.

Ví dụ:

LASTZERO.INP	LASTZERO.OUT
3 2 50 4 20	3
5 3 15 16 3 25 9	3
3 3 9 77 13	0

Ghi chú:

<u>Sub 1</u>: 60% số điểm có n ≤ 20.

<u>Sub 2</u>: 40% số điểm có n ≤ 100.

Hưởng dẫn giải thuật: LASTZERO

Ta có nhận xét số chữ số 0 tận cùng bên phải của một số bằng min(pw2, pw5) trong đó pw2 và pw5 là số lượng thừa số 2 và 5 trong dạng phân tích thành nhân tử của số đó. Vậy trước tiên, ta sẽ tính $pw2_i$ và $pw5_i$ của mỗi a_i.

Sub 1: Quay lui sinh 2^k dãy nhị phân rồi duyệt bình thường.

Sub2: Ta sẽ sử dụng thuật toán quy hoạch động như sau:

Hàm đệ quy có nhớ solve(i, j, l) trong đó i kiểm soát các phần tử của dãy a, j kiểm soát số lượng phần tửđãchọnvà l kiểmsoát pw5 củatíchcácphầntửđãchọn. Hàmsẽtrảvề pw2 lớnnhấtcóthểcủatíchcácphầntửđượcchọn.

Côngthức:

solve(i, j, l) = max(solve(i, j, l), solve(i + 1, j, l)) (Khôngchọna_i)

solve(i, j, l) = max(solve(i, j, l), solve(i + 1, j + 1, l + $pw5_i$) + $pw2_i$) (Chọna_i)

Vì n ≤ 100

nênviệccàiđặtthuậttoántheomôhìnhnhưtrênsẽbịtràn bộnhớmặcdùvẫnđủđảmbảovềmặtthờigian. Ta sẽkhửđệquybằngcáchdùngmảng f:

f[i + 1][j][l] = max(f[i + 1][j][l], f[i][j][l])

f[i + 1][j + 1][l + $pw5_i$] = max(f[i + 1][j + 1][l + $pw5_i$], f[i][j][l] + $pw2_i$)

Đápáncủabàitoán: max(min(l, f[n + 1][k][l]))

Nhậnthấy f[i + 1][…][…] chỉcậpnhậtkếtquảtheo f[i][…][…] nên ta cóthểkhử 1 chiềucủa f bằngcáchsửdụng 2 mảng2 chiềutínhtoán song hànhvớinhau, vậylàgiảiquyếtđượcvấnđềvềbộnhớ.

Bài 3: **Cuộc đua xe F1**

Một cuộc thi đua xe công thức 1 mở rộng vô cùng hoành tráng sắp được tổ chức tại thành phố XYZ. Ray cũng sẽ tham gia cuộc đua xe lần này. Vốn là một tay đua xe có hạng, cậu rất tự tin rằng không ai có thể đánh bại được mình trong bất kì cuộc đua nào. Tuy nhiên, thể lệ cuộc đua xe lần này lại có nhiều điều đặc biệt khiến cậu cảm thấy rất bối rối. Như mọi lần khác, cuộc đua được tổ chức trên quy mô toàn thành phố, mỗi thí sinh sẽ chạy xe theo các con đường được kẻ vạch sẵn để di chuyển qua những địa điểm dừng. Có tất cả n địa điểm dừng được đánh số thứ tự từ 1 đến n, 2 địa điểm bất kì đều có 2 con đường kết nối. Điểm đặc biệt của cuộc thi lần này là mỗi thí sinh sẽ phải sử dụng những chiếc xe đua được ban tổ chức chuẩn bị sẵn cho từng người. Cụ thể, mỗi thí sinh được chuẩn bị sẵn m chiếc xe khác nhau và có thể đổi xe liên tục tại bất kì địa điểm dừng nào (mỗi chiếc xe có thể sử dụng lại nhiều lần, thời gian đổi xe không đáng kể). Một chiếc xe lại tốn một khoảng thời gian trung bình khác nhau để di chuyển qua mỗi đoạn đường khác nhau. Cuộc thi gồm có r vòng đua, tại vòng thứ i ban tổ chức lại yêu cầu các thí sinh xuất phát từ địa điểm s_i và đích đến là địa điểm f_i, đồng thời tại vòng này các thí sinh chỉ được phép đổi xe tối đa k_i lần. Ray đua xe rất giỏi nhưng thể lệ cuộc đua lần này quá mới lạ với cậu, vì thế nên Ray cần tới sự giúp đỡ của bạn. Hãy tính toán cách đua xe hợp lý để Ray có thể hoàn thành mỗi vòng đua trong tổng thời gian nhỏ nhất với mỗi vòng nhé!

Dữ liệu: Vào từ file văn bản RACE.INP gồm:

-Dòng thứ nhất gồm 3 số nguyên dương n, m và r ($2 \le n \le 50$).

-Tiếp theo là m bảng 2 chiều kích thước n x n biểu diễn thông tin của những chiếc xe, trong mỗi bảng ô giao giữa dòng i và cột j gồm 1 số nguyên dương a_{ij} ($0 \le a_{ij} \le 10^6$) là thời gian để chiếc xe tương ứng di chuyển qua đoạn đường nối 2 địa điểm dừng i, j.

-r dòng tiếp theo, dòng thứ i gồm 3 số nguyên s_i, t_i, k_i ($1 \le s_i, f_i \le n$; $s_i \ne f_i$; $0 \le k_i \le 10^5$) cho biết địa điểm xuất phát, địa điểm đích và số lần đổi xe tối đa của vòng đua thứ i.

Kết quả: Đưa ra file văn bản RACE.OUT gồm r dòng, mỗi dòng ghi tổng thời gian nhỏ nhất cần để Ray hoàn thành vòng đua tương ứng.

Ví dụ:

RACE.INP	RACE.OUT
4 2 3 0 1 5 6 2 0 3 6 1 3 0 1 6 6 7 0 0 3 5 6 2 0 1 6 1 3 0 2	3 4 3

6 6 7 0 1 4 2 1 4 1 1 4 3	

Ghi chú:

<u>Sub 1</u>: 30% số điểm có m = 1, $1 \le r \le 10^5$.

<u>Sub 2</u>: 30% số điểm có $1 \le m \le 50$, $1 \le r \le 10^5$, $k_i = 0$.

<u>Sub 3</u>: 40% số điểm có $1 \le m \le 50$, $1 \le r \le 10^5$.

Hướng dẫn bài RACE

Trướctiên, ta sẽdùngthuậttoán Floyd tìmđườngđingắnnhấtgiữamọicặpđịađiểmđốivớimỗichiếcxe.Lưulạikếtquảvàomảng d, trongđó d[i][u][v] làđộdàiđườngđingắnnhấttừđịađiểm u đếnđịađiểm v nếusửdụngxethứ i.

Sub 1: Vì m = 1 tứclà ta chỉcóduynhất 1 chiếcxeđểsửdụng, khiđó ở vòngđuathứ i ta khôngcầnquantâmđếnk_inữa. Đápánlà$d[1][s_i][f_i]$.

Sub 2: Tạivòngđuathứ i, vì$k_i = 0$ nên ta chỉđượcsửdụng 1 chiếcxe, ta sẽchọnchiếctốiưunhất. Đápánlà$\min(d[j][s_i][f_i])$.

Sub 3: Gọi f[u][v][l] làđộdàiđườngđingắnnhấttừđịađiểm u đếnđịađiểm v vớisốlầnđổixetốiđalà l. Từ sub 2 ta suyra f[u][v][0] = min(d[i][u][v]). Dựavàođó, ta dễdàngtìmracôngthức QHĐ tínhtoánmảng f:

f[u][v][l] = min(f[u][x][l – 1] + f[x][v][0])

Tuynhiênvấnđề ở đây$k_i \le 10^5$làrấtlớn, nếu ta QHĐ đơnthuầntheocôngthứctrênthìsẽbịquáthờigian. Nhậnxétlàđườngđingắnnhất ta cầntìmluônlàđườngđiđơn, tứclàmỗiđịađiểmxuấthiệntrênđườngđikhôngquá 1 lần.Mà ta lạichỉcóthểđổixemỗikhidừnglạitạiđịađiểmnàođó, vậynênthựcchấtsốlầntốiđamà ta cầnđổixekhôngbaogiờvượtquá n.

Bài 4: Dãy số

Trong giờ tự học, Norman đã đưa ra một bài toán rất khó để thách thức Emma và những đứa trẻ khác. Cậu cho trước một dãy số nguyên gồm n phần tử và yêu cầu mọi người tìm cách đưa giá trị của tất cả các phần tử này về cùng bằng một số h. Cậu cho phép mọi người thực hiện vô số bước, mỗi bước được chọn một đoạn con liên tiếp [l, r] và tăng tất cả phần tử trong đoạn lên 1 đơn vị. Tuy nhiên, Norman đưa ra luật là đoạn con được chọn phải không được trùng vị trí đầu hay vị trí cuối với bất kì đoạn con nào được chọn trước đó. Nói cách khác, mỗi cặp đoạn con được chọn [l1, r1], [l2, r2] phải thỏa mãn $l1 \ne l2$ và $r1 \ne r2$. Emma rất nhanh trí và thông minh nên đã dễ dàng tìm được một lời giải thỏa mãn yêu cầu đề bài. Tưởng chừng bài toán của Norman chỉ dễ dàng xơi thế nhưng rồi Norman yêu cầu phải đưa ra tổng số lượng cách giải khác nhau. Emma đã cố gắng mò mẫn tính toán nhưng kết quả bài toán quá lớn, đành phải nhờ đến khả năng lập trình của các Coders thông thái thôi!

Dữ liệu: Vào từ file văn bản SEQUENCE.INP:

-Dòng đầu tiên gồm hai số nguyên dương n và h ($h \le 2000$).

-Dòng thứ hai gồm n số nguyên $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 2000$).

Kết quả: Đưa ra file văn bản SEQUENCE.OUT: gồm một số nguyên duy nhất là phần dư kết quả bài toán khi chia cho $10^9 + 7$.

Ví dụ:

SEQUENCE.INP	SEQUENCE.OUT
3 2	4

1 1 1	

Ghi chú:

<u>Sub 1</u>: 50 % số điểm có n ≤ 10.

<u>Sub 2</u>: 50 % số điểm có n ≤ 2000.

SEQUENCE

Sub 1: Vì n nhỏnên ta cóthểduyệtbằngđệquy.

Sub 2: Ta sẽsửdụngthuậttoán QHĐ theomôhìnhđệquycónhớ:

solve(pos, open) trongđóposkiểmsoátcácphầntửcủamảng a, open kiểmsoátsốlượngngoặcmởchưađượcghépcặpvớingoặcđónghiệntại.

Hàmtrảlạitổngsốcáchgiảibàitoáncủa Norman.Côngthức QHĐ:

-Nếua[pos] + open = h - 1: (lúcnày ta bắtbuộcphảităng a[pos] lên 1 đơnvị)

+) solve(pos, open) += solve(pos + 1, open) (tạo 1 cặpngoặcmởđóng ngaytạipos)

+) solve(pos, open) += solve(pos + 1, open + 1) (tạothêm 1 ngoặcmởtại pos)

+) solve(pos, open) += open * solve(pos + 1, open) (tạothêm 1 cặpngoặc đóngmởtạipos, ngoặcđóngnàycóthểghépcặpvới open ngoặcmở trướcđó)

-Nếua[pos] + open = h:

+) solve(pos, open) += solve(pos + 1, open) (bỏ qua pos, vềsauposs̃ đượcđưavàogiữacặpngoặcmởđóngnàođó)

+) solve(pos, open) += open * solve(pos + 1, open - 1) (tạo 1 ngoặcđóngtại pos, ngoặcđóngnàycóthểghépcặpvới open ngoặcmởtrướcđó)

Neo đệquy: nếupos = n + 1, kiểmtraxem open cóbằng 0 hay không. Nếu open = 0 thìtrảvề 1, ngượclạitrảvề 0.