

Hệ thống

Alice đang xây dựng một hệ thống được đặc trưng bằng một dãy gồm n số nguyên dương $(a_1, a_2, \dots, a_n)$. Với hai chỉ số $1 \leq L < R \leq n$, Alice định nghĩa độ ổn định đoạn $[L, R]$ của hệ thống bằng tổng tất cả tích các cặp số trong đoạn, cụ thể: $S(L, R) = \sum_{L \leq i < j \leq R} a_i \times a_j$.

Ví dụ, dãy $(2, 1, 3, 5)$, ta có $S(1, 2) = 2 \times 1 = 2$; $S(2, 4) = 1 \times 3 + 1 \times 5 + 3 \times 5 = 23$.

Trong quá trình vận hành hệ thống, một số đặc trưng bị thay đổi và Alice cần phải tính độ ổn định của một số đoạn. Cụ thể, sẽ có q sự kiện lần lượt xảy ra, mỗi sự kiện thuộc một trong hai loại sau:

- Loại 1 có dạng: $1 \ i \ x$, trong đó $1 \leq i \leq n$ và $1 \leq x \leq 10^6$, có nghĩa là số thứ i của dãy bị thay đổi thành x .
- Loại 2 có dạng: $2 \ L \ R$, trong đó $1 \leq L < R \leq n$, có nghĩa là cần tính độ ổn định đoạn $[L, R]$ của hệ thống.

Yêu cầu: Cho dãy $(a_1, a_2, \dots, a_n)$ và q sự kiện, với mỗi sự kiện loại 2 hãy đưa ra độ ổn định của đoạn cần tính.

Dữ liệu: Vào từ file văn bản **STSYS.INP**:

- Dòng đầu tiên chứa hai số nguyên dương n, q ($2 \leq n, q \leq 2 \times 10^5$).
- Dòng thứ hai chứa n số nguyên dương $a_1, a_2, \dots, a_n$ ($a_i \leq 10^6$).
- Tiếp theo là q dòng, mỗi dòng chứa ba số $1 \ i \ x$ hoặc $2 \ L \ R$ mô tả loại dạng sự kiện xảy ra.

Kết quả: Ghi ra file văn bản **STSYS.OUT** gồm một số dòng, mỗi dòng chứa một số là độ ổn định cần tính tương ứng với sự kiện loại 2 xảy ra. Vì giá trị này có thể rất lớn nên chỉ cần đưa ra giá trị $S(L, R) \% (10^9 + 7)$, trong đó phép toán $\%$ là phép toán chia lấy dư.

Ràng buộc:

- Có 40% số test thỏa mãn: $n, q \leq 200$.
- 30% số test khác thỏa mãn: $a_i \leq 10^3$ và cả q sự kiện đều là thao tác loại 2.
- 30% số test còn lại không có ràng buộc nào thêm.

Ví dụ:

STSYS.INP	STSYS.OUT
4 4	2
2 1 3 5	23
2 1 2	23
2 2 4	
1 1 5	
2 1 3	

Chuỗi DNA

Alice đang nghiên cứu về mã di truyền, cô cần tạo ra các chuỗi DNA được biểu diễn bằng xâu nhị phân. Có một số mẫu nhị phân ngắn, được gọi là “**mảnh vỡ cảm**” có tính chất không ổn

định, nếu chúng xuất hiện trong chuỗi DNA như là một xâu con liên tiếp, chúng sẽ gây ra một phản ứng dây chuyền phá hủy chuỗi.

Alice đã xác định được k mảnh vỡ cắm $p_1, p_2, \dots, p_k$. Để đảm bảo sự ổn định, Alice phải tạo ra các chuỗi DNA nhị phân có độ dài đúng bằng n mà **tránh hoàn toàn** tất cả k mảnh vỡ cắm.

Yêu cầu: Cho n và các xâu nhị phân $p_1, p_2, \dots, p_k$, hãy giúp Alice đếm số lượng chuỗi DNA nhị phân tránh tất cả k mảnh vỡ cắm.

Dữ liệu: Vào từ file văn bản **DNASEQ.INP**:

- Dòng đầu chứa hai số nguyên n và k ($n \leq 200; k \leq 10$);
- Dòng thứ i ($1 \leq i \leq k$) trong k dòng tiếp theo chứa xâu nhị phân p_i (độ dài các xâu không vượt quá n).

Kết quả: Ghi ra file văn bản **DNASEQ.OUT** một số nguyên là giá trị $e \% 111539786$, trong đó e là số chuỗi DNA nhị phân thỏa mãn và phép toán $\%$ là phép toán chia lấy dư.

Ràng buộc:

- Có 40% số test thỏa mãn: $n \leq 20$.
- 30% số test khác thỏa mãn: $k = 1$.
- 30% số test còn lại không có ràng buộc nào thêm.

Ví dụ:

DNASEQ.INP	DNASEQ.OUT	DNASEQ.INP	DNASEQ.OUT
2 1 0	1	2 2 00 10	2

Jump

Trong rừng mưa nhiệt đới Sumatra, có N cây liên tiếp nhau trên một hàng được đánh số từ 0 đến $N - 1$ từ trái sang phải. Tất cả các cây có chiều cao đôi một khác nhau, cây i có chiều cao $H[i]$. Pak Dengklek đang huấn luyện một con đười ươi nhảy từ cây này sang cây khác. Trong một lần nhảy, đười ươi có thể nhảy từ ngọn cây này sang ngọn cây gần nhất, bên trái hoặc bên phải, mà có chiều cao lớn hơn cái cây nó đang bám. Cụ thể, nếu đười ươi hiện đang ở cây x , thì nó có thể nhảy sang cây y khi và chỉ khi thỏa mãn một trong hai điều sau:

- y là số nguyên không âm lớn nhất nhỏ hơn x mà $H[y] > H[x]$;
- y là số nguyên không âm nhỏ nhất lớn hơn x mà $H[y] > H[x]$.

Pak Dengklek có Q phương án huấn luyện cho đười ươi nhảy, mỗi phương án có thể được biểu diễn dưới dạng bốn số nguyên A, B, C và D ($A \leq B < C \leq D$). Đối với mỗi phương án, Pak Dengklek muốn biết liệu con đười ươi có thể bắt đầu từ cây s nào đó ($A \leq s \leq B$) và kết thúc ở cây e nào đó ($C \leq e \leq D$) bằng cách sử dụng một chuỗi các bước nhảy hay không. Nếu có thể, Pak Dengklek muốn biết số lượng bước nhảy ít nhất mà đười ươi cần sử dụng cho phương án đó.

Input

- dòng 1: N Q ($2 \leq N \leq 200\,000$, $1 \leq Q \leq 100\,000$)
- dòng 2: $H[0]$ $H[1]$... $H[N-1]$
- dòng $3 + i$ ($0 \leq i \leq Q-1$): A B C D tương ứng với phương án huấn luyện thứ i .

Output

- dòng $1 + i$ ($0 \leq i \leq Q-1$): trả về giá trị của phương án huấn luyện thứ i .

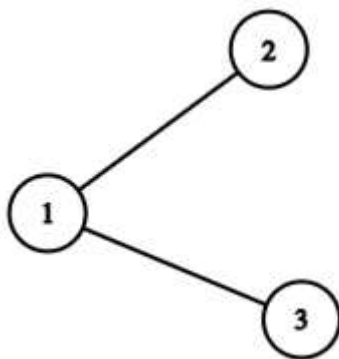
Subtask

- $H[i] = i + 1$ (với mọi $0 \leq i \leq N-1$)
- $N \leq 200$, $Q \leq 200$
- $N \leq 2000$, $Q \leq 2000$
- $Q \leq 5$
- $A = B$, $C = D$
- $C = D$
- Không có ràng buộc nào thêm.

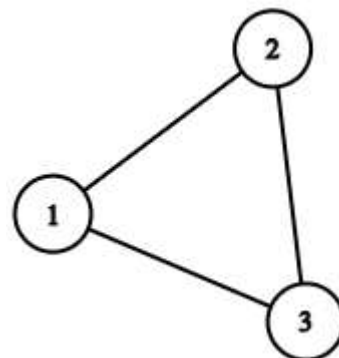
Input	Output
7 2	2
3 2 1 6 4 5 7	1
4 4 6 6	
1 3 5 6	

Tạo đồ thị

Alice cần tạo một đồ thị gồm n đỉnh, các đỉnh được đánh số từ 1 đến n , có các tính chất: Vô hướng; Không có khuyên (không có cạnh nối chính nó); Không có cạnh bội (giữa hai đỉnh chỉ có nhiều nhất một cạnh); Liên thông (từ một đỉnh bất kì đều có đường đi tới một đỉnh bất kì khác); Đỉnh i ($1 \leq i \leq n$) có bậc là d_i .



Đồ thị với dãy bậc $(d_1, d_2, d_3) = (2, 1, 1)$



Đồ thị với dãy bậc $(d_1, d_2, d_3) = (2, 2, 2)$

Yêu cầu: Cho dãy số nguyên dương $d_1, d_2, \dots, d_n$, hãy giúp Alice xây dựng đồ thị thỏa mãn.

Dữ liệu: Vào từ file văn bản **MKGRAPH.INP**:

- Dòng đầu chứa số nguyên n ($n \leq 10^4$);

- Dòng thứ hai chứa n số nguyên $d_1, d_2, \dots, d_n$ ($0 < d_i < n; d_1 + d_2 + \dots + d_n \leq 10^5$).
Dữ liệu bảo đảm tồn tại đồ thị thỏa mãn.

Kết quả: Ghi ra file văn bản **MKGRAPH.OUT** gồm một số dòng, mỗi dòng chứa hai số i, j mô tả một cạnh của đồ thị.

Ràng buộc:

- Có 30% số test của bài có $n \leq 5$;
- Có 20% số test khác của bài có tính chất: $n \leq 100$ và đồ thị cần xây dựng có dạng cây;
- Có 10% số test khác của bài có tính chất: Đồ thị cần xây dựng có dạng cây;
- Có 20% số test khác của bài có tính chất: $n \leq 100$;
- Có 20% số test còn lại của bài không có ràng buộc nào thêm.

Ví dụ:

MKGRAPH.INP	MKGRAPH.OUT
3 2 1 1	1 2 1 3
3 2 2 2	1 2 1 3 2 3

Thiết kế mê cung (mazegen)

Alice đang xây dựng trò chơi tìm đường trong mê cung, trong đó mê cung được biểu diễn bằng một lưới kích thước $m \times n$ ô. Các hàng của lưới được đánh số từ 1 đến m từ trên xuống, các cột của lưới được đánh số từ 1 đến n từ trái sang phải. Ô nằm ở hàng i ($1 \leq i \leq m$), cột j ($1 \leq j \leq n$) được gọi là ô (i, j) , ô (i, j) có thể là ô tự do đi vào được hoặc là ô tường không thể đi vào được. Người chơi xuất phát tại ô tự do $(1, 1)$ và cần di chuyển nhanh nhất để đến ô tự do (m, n) bằng cách di chuyển qua các ô tự do chung cạnh.

Alice đã thiết kế được một mê cung và người chơi cần di chuyển qua ít nhất d ô (bao gồm cả ô xuất phát và ô kết thúc), nhưng do sơ ý, cô đã đánh mất dữ liệu của mê cung. Cô chỉ nhớ một số ô là tường nên hiện tại cô có một mê cung với các ô tường là các ô mà cô nhớ. Hãy giúp Alice tạo ra được một mê cung bằng cách thay một số ô tự do thành ô tường để khi đó số lượng ô trên đường đi ngắn nhất từ ô $(1, 1)$ đến ô (m, n) gần giá trị d nhất.

Input

- Dòng đầu chứa ba số nguyên dương m, n, d ($m, n \leq 30$);
- Dòng thứ i ($1 \leq i \leq m$) trong m dòng sau, mỗi dòng chứa một xâu độ dài n chỉ gồm hai loại kí tự '.' hoặc '#' tương ứng mô tả là ô tự do hay ô tường.
Dữ liệu đảm bảo có thể tạo được mê cung để đường đi ngắn nhất từ ô $(1, 1)$ đến ô (m, n) qua đúng d ô.

Output

- Gồm m dòng, mỗi dòng chứa một xâu độ dài n chỉ gồm hai loại kí tự '.' hoặc '#' mô tả mê cung.

Input	Output
5 4 14	
....	###.
....	
....	.###
.###	
....	

Chấm điểm

Với mỗi test, gọi s là số lượng ô trên đường đi ngắn nhất từ ô $(1,1)$ đến ô (m,n) trên mê cung thí sinh tạo ra, đặt $x = |s - d|$, khi đó, thí sinh sẽ đạt: $\max\{100\% - x \times 10\%, 0\}$ số điểm của test đó.

Subtask 1 (20%): $m = 2$.

Subtask 2 (20%): $m = 3$.

Subtask 3 (60%): Không có ràng buộc nào thêm.