

Đề Thi Thử VOI2024-2025

Ngày 2

Thời gian làm bài: 180 phút

TỔNG QUAN ĐỀ THI

Tên bài	Tên chương trình	Dữ liệu đầu vào	Kết quả đầu ra	Số điểm
Bài 4. Phép nén đồ thị	GRAPHZIP.*	GRAPHZIP.INP	GRAPHZIP.OUT	7 điểm
Bài 5. Đồ thị bội chung nhỏ nhất	LCMGRAPH.*	LCMGRAPH.INP	LCMGRAPH.OUT	7 điểm
Bài 6. Hàng rào hoàn hảo	ROBOT.*	ROBOT.INP	ROBOT.OUT	6 điểm

Dấu * được thay thế bởi py hoặc cpp của ngôn ngữ lập trình tương ứng là Python và C++.

Bài 4. GRAPHZIP - Phép nén đồ thị

Hãy làm trước giải các bài toán sau:

Cho một đồ thị G gồm n đỉnh và m cạnh hai chiều. Các đỉnh được đánh số từ 1 đến n , các cạnh được đánh số từ 1 đến m . Cạnh thứ i nối giữa hai đỉnh a_i và b_i và có trọng số w_i . Lưu ý rằng, đồ thị G có thể là đa đồ thị (có cạnh nối giữa một đỉnh với chính nó, hoặc có nhiều hơn hai cạnh nối cùng một cặp đỉnh).

Giáo sư X vừa phát minh ra một phép toán để nén đồ thị G bên trên như sau:

- Trước hết, cần tô màu cho mỗi đỉnh trong đồ thị G bằng một trong số các màu từ 1 đến k (các đỉnh có thể có màu khác nhau).
- Sau đó, xây dựng đồ thị G' gồm k đỉnh và m cạnh, trong đó cạnh thứ i nối giữa hai đỉnh x và y và có trọng số w_i (với x và y lần lượt là màu của hai đỉnh của a_i và b_i).
- Đồ thị G' là kết quả của phép nén đồ thị G .

Hiện tại, đã có một số đỉnh của đồ thị G được tô màu. Cụ thể, đỉnh i đã được tô màu c_i (nếu $c_i > 0$) hoặc chưa được tô màu nếu $c_i = 0$. Giáo sư X đưa ra định nghĩa về cây khung nhỏ nhất như sau:

- Cây khung của đồ thị là một tập hợp các cạnh của đồ thị thỏa mãn tập cạnh này không chứa chu trình và liên thông (từ một đỉnh bất kỳ có thể đi tới bất kỳ đỉnh nào khác theo mà chỉ dùng các cạnh trên cây khung).
- Trong đồ thị có trọng số, cây khung nhỏ nhất là cây khung có tổng trọng số các cạnh trong cây nhỏ nhất (giá trị tổng này được gọi là trọng số của cây khung).

Yêu cầu: Hãy giúp giáo sư X tô màu các đỉnh còn lại của đồ thị G sao cho trọng số của cây khung nhỏ nhất của đồ thị G' là nhỏ nhất có thể, hoặc cho biết rằng không tồn tại cách tô màu để đồ thị G' có chứa cây khung.

Dữ liệu

- Dòng đầu tiên gồm số nguyên t ($1 \leq t \leq 10$) là số lượng test case. Các test case được cách nhau bằng một dòng trống. Mỗi test case được mô tả như sau.

- Dòng đầu tiên gồm ba số nguyên n, m, k ($1 \leq n, m \leq 10^5, 1 \leq k \leq n$) là số đỉnh, số cạnh của đồ thị G , và số màu khác nhau có thể tô cho từng đỉnh.
- Dòng thứ hai gồm n số nguyên $c_1, c_2, \dots, c_n$ ($0 \leq c_i \leq n$) là màu được tô ban đầu của đỉnh i (nếu $c_i > 0$) hoặc cho biết đỉnh i chưa được tô màu (nếu $c_i = 0$).
- Dòng thứ i trong m dòng tiếp theo gồm ba số nguyên a_i, b_i, w_i ($1 \leq a_i, b_i \leq n, 1 \leq w_i \leq 10^9$) mô tả cạnh thứ i của đồ thị G . Lưu ý rằng, đồ thị G có thể là đa đồ thị.

Dữ liệu vào đảm bảo rằng tổng n của các test case không vượt quá 3×10^5 , và tổng m của các test case không vượt quá 3×10^5 .

Kết quả

Với mỗi test case, in ra một số nguyên duy nhất là trọng số nhỏ nhất có thể của cây khung nhỏ nhất của đồ thị G' với cách tô màu tối ưu, hoặc in ra -1 nếu không tồn tại cách tô màu để đồ thị G' có chứa cây khung.

Ràng buộc

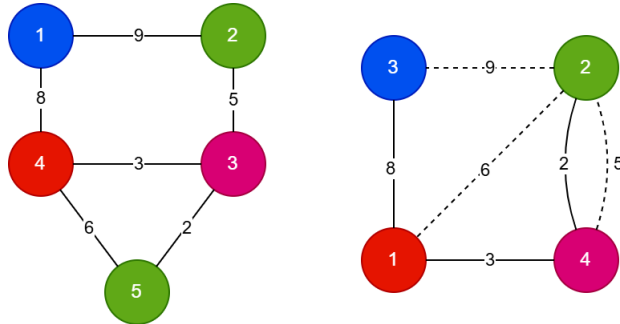
- Có 20% số điểm thỏa mãn: $n = 2, m = 1$
- 20% số điểm tiếp theo thỏa mãn: $c_i = i$
- 20% số điểm tiếp theo thỏa mãn: $n \leq 6, m \leq 20$
- 20% số điểm tiếp theo thỏa mãn: $c_i = 0$
- 20% số điểm còn lại không có ràng buộc gì thêm

Ví dụ

GRAPHZIP.INP	GRAPHZIP.OUT
4	13
5 6 4	2
3 2 4 1 2	-1
1 2 9	10
2 3 5	
1 4 8	
5 4 6	
3 4 3	
3 5 2	
3 3 2	
1 2 0	
1 2 5	
1 3 3	
2 3 2	
4 1 3	
0 0 0 0	
1 2 7	
3 3 2	
0 2 0	
1 2 10	
1 2 20	
3 3 30	

Lưu ý

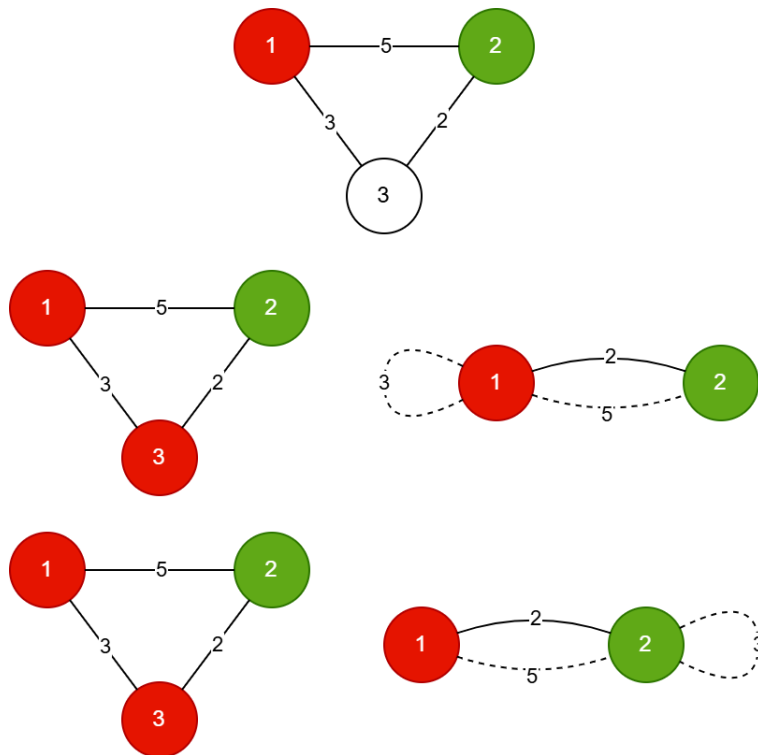
Với test case thứ nhất, các đỉnh đều đã được tô màu từ trước. Cây khung nhỏ nhất của đồ thị G' gồm các cạnh $(1, 3, 8)$, $(1, 4, 3)$, $(2, 4, 2)$ với tổng trọng số $8 + 3 + 2 = 13$.



Hình vẽ minh họa test case thứ nhất. Bên trái là đồ thị G , bên phải là đồ thị G' .

Với test case thứ hai, đỉnh 3 là đỉnh duy nhất chưa được tô màu.

- Nếu tô đỉnh 3 bằng màu 1 thì cây khung nhỏ nhất của đồ thị G' sẽ có trọng số 3.
- Nếu tô đỉnh 3 bằng màu 2 thì cây khung nhỏ nhất của đồ thị G' sẽ có trọng số 2.



Hình vẽ minh họa test case thứ hai.

Với test case thứ ba, đồ thị G' có 3 đỉnh nhưng chỉ có 1 cạnh nên sẽ không tồn tại cây khung với mọi cách tô màu đồ thị G .

Với test case thứ tư, lưu ý rằng đồ thị G có thể là đa đồ thị.

Bài 5. Đồ thị bội chung nhỏ nhất

Phong là một người yêu thích các bài toán liên quan đến đồ thị và số học. Một ngày nọ, anh phát hiện ra một bài toán thú vị trong khi làm việc với một mảng số nguyên và một đồ thị đặc biệt.

Từ một mảng $a_1, a_2, \dots, a_n$, Phong xây dựng ra đồ thị vô hướng đầy đủ G với n đỉnh được đánh số từ 1 đến n , trong đó giữa hai đỉnh i và j ($i < j$) có một cạnh vô hướng kết nối chúng với trọng số là $\text{LCM}(A_i, A_{i+1}, \dots, A_j)$.

Nhắc lại, $\text{LCM}(A_i, A_{i+1}, \dots, A_j)$ là số nguyên dương nhỏ nhất chia hết cho $A_i, A_{i+1}, \dots, A_j$. Ví dụ $\text{LCM}(4, 6) = 12$ hay $\text{LCM}(3, 7, 14) = 42$.

Phong đánh dấu k đỉnh khác nhau $p_1, p_2, \dots, p_k$ trên đồ thị G làm các *đỉnh đặc biệt*. Anh muốn chọn một số cạnh trong G sao cho mỗi cặp *đỉnh đặc biệt* đều được kết nối với nhau. Hai đỉnh u và v được xem là kết nối với nhau nếu tồn tại một đường đi ($u = x_0, x_1, x_2, \dots, x_d = v$) giữa hai đỉnh đó, sao cho với mỗi cặp đỉnh liên tiếp (x_{i-1}, x_i) ($1 \leq i \leq d$) được kết nối trực tiếp với nhau bằng một cạnh đã chọn.

Vì có nhiều cách chọn, Phong muốn tổng trọng số của các cạnh đã chọn là nhỏ nhất.

Yêu cầu: Bạn hãy giúp Phong tìm ra tổng trọng số nhỏ nhất đó!

Dữ liệu

- Dòng đầu tiên chứa hai số nguyên dương n, k ($1 \leq k \leq n \leq 3 \cdot 10^5$) lần lượt là số đỉnh của đồ thị G và số đỉnh đặc biệt.
- Dòng tiếp theo chứa n số nguyên $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 3 \cdot 10^5$) mô tả mảng a .
- Dòng cuối cùng chứa k số nguyên dương phân biệt $p_1, p_2, \dots, p_k$ ($1 \leq p_i \leq n$) mô tả dãy p — các đỉnh đặc biệt được chọn.

Kết quả

- Dòng đầu tiên chứa một số nguyên duy nhất t ($0 \leq t \leq 10^{18}$) là tổng trọng số nhỏ nhất trong phương án tối ưu.
- Dòng thứ hai chứa một số nguyên m ($0 \leq m \leq 10^6$) mô tả số cạnh được chọn trong phương án tối ưu.
- m dòng tiếp theo mỗi dòng chứa hai số nguyên u, v ($1 \leq u, v \leq n, u \neq v$) mô tả một cạnh được chọn. Các cạnh này có thể xuất hiện theo bất kỳ thứ tự nào, và mỗi cạnh chỉ xuất hiện tối đa một lần.

Nếu có nhiều hơn một phương án tối ưu, in ra bất kỳ. Có thể chứng minh được rằng luôn tồn tại phương án tối ưu thỏa mãn các điều kiện ở trên.

Ràng buộc

- Có 30% số test ứng với 30% số điểm của bài có $n \leq 6$.
- 20% số test khác ứng với 20% số điểm của bài có $n \leq 200$.
- 20% số test khác ứng với 20% số điểm của bài có $n \leq 2000$.
- 15% số test khác ứng với 15% số điểm của bài có $a_i \leq 16$ với mọi $1 \leq i \leq n$.
- 15% số test còn lại không có ràng buộc gì thêm.

Ngoài ra, với mỗi test, bạn nhận được 60% số điểm của test đó nếu bạn cho biết chính xác tổng trọng số tối thiểu mà không cho biết phương án chọn cạnh để đạt được tổng trọng số đó, hoặc nếu phương án bạn đưa ra không hợp lệ, không khớp với tổng trọng số đã tính toán, hoặc không thỏa mãn điều kiện,...

Ví dụ

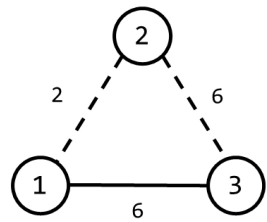
LCMGRAPH.INP	LCMGRAPH.OUT
3 2 1 2 3 1 3	6 1 3 1
8 4 6 3 2 3 1 10 3 15 6 8 2 7	61 4 2 5 7 8 5 6 5 8

Lưu ý

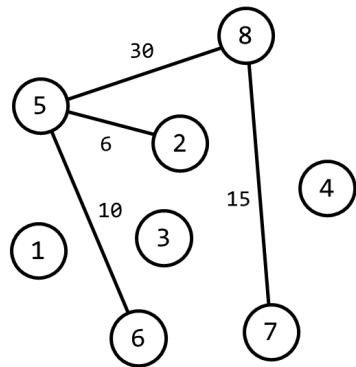
Ở ví dụ thứ nhất, ta có dãy $a = (1, 2, 3)$, do đó đồ thị G có 3 đỉnh với các cạnh có trọng số lần lượt là:

- Cạnh $(1, 2)$ có trọng số $\text{LCM}(a_1, a_2) = \text{LCM}(1, 2) = 2$.
- Cạnh $(2, 3)$ có trọng số $\text{LCM}(a_2, a_3) = \text{LCM}(2, 3) = 6$.
- Cạnh $(1, 3)$ có trọng số $\text{LCM}(a_1, a_2, a_3) = \text{LCM}(1, 2, 3) = 6$.

Với các *đỉnh đặc biệt* là 1 và 3, để kết nối chúng với trọng số nhỏ nhất, ta chọn cạnh $(1, 3)$ với tổng trọng số là 6. Đây là cách có tổng trọng số nhỏ nhất. Bạn có thể tham khảo hình bên dưới, trong đó các cạnh *nét liền* biểu thị các cạnh được chọn trong phương án tối ưu, các cạnh *nét đứt* biểu thị cạnh không được chọn.



Minh họa cho ví dụ thứ nhất



Minh họa cho ví dụ thứ hai

Bài 6. Hàng rào hoàn hảo

Trước bức tường cao của một phòng thí nghiệm bí mật, có N cảm biến có khả năng nhận diện những kẻ xâm nhập. Bức tường được biểu diễn bằng một đoạn thẳng $[0, L]$, và các cảm biến ban đầu được đặt tại một số điểm nằm trên đoạn này.

Mỗi cảm biến có **phạm vi nhận diện** r , và phạm vi này giống nhau với tất cả các cảm biến. Nói cách khác, một cảm biến tại vị trí p có thể nhận diện kẻ xâm nhập trong khoảng $[p - r, p + r]$. Khoảng này được gọi là **phạm vi nhận diện của cảm biến**.

Một robot duy nhất có nhiệm vụ quản lý biên giới của bức tường. Ban đầu, robot được đặt tại **đầu bên trái của bức tường** (nghĩa là điểm 0) và không mang theo cảm biến nào. Robot có thể di chuyển sang trái hoặc phải dọc theo tường, đồng thời tháo lắp và cầm theo cảm biến. Robot chỉ có thể tháo hoặc lắp cảm biến tại vị trí bất kỳ trong đoạn $[0, L]$. Số lượng cảm biến mà robot có thể mang theo cùng lúc là không giới hạn.

Nếu hợp của tất cả các phạm vi nhận diện của cảm biến bao phủ hoàn toàn đoạn tường $[0, L]$, các cảm biến sẽ tạo ra một **hàng rào bảo vệ hoàn hảo**. Robot phải dời vị trí của các cảm biến nếu cần thiết để đạt được hàng rào bảo vệ hoàn hảo, đồng thời tối thiểu hóa tổng quãng đường di chuyển của robot.

Yêu cầu: Hãy viết chương trình tính **tổng quãng đường di chuyển tối thiểu** mà robot phải thực hiện để tạo ra hàng rào bảo vệ hoàn hảo, dựa trên độ dài đoạn tường $[0, L]$, vị trí ban đầu của N cảm biến, và phạm vi nhận diện r của các cảm biến.

Ràng buộc:

- $1 \leq N \leq 1,000,000$
- $1 \leq L \leq 10^{12}$
- $1 \leq r \leq 10^{12}$
- Cả L và r đều là số nguyên.
- Tất cả các vị trí ban đầu của cảm biến là số nguyên nằm trong khoảng $[0, L]$.
- Vị trí ban đầu của các cảm biến được cho theo thứ tự không giảm.

Dữ liệu

Dòng đầu tiên chứa ba số nguyên N , L , và r , lần lượt biểu diễn số lượng cảm biến, chiều dài của bức tường, và phạm vi nhận diện của cảm biến.

Dòng thứ hai chứa N số nguyên, biểu diễn vị trí ban đầu của các cảm biến, được cho theo thứ tự không giảm.

Kết quả

Nếu **không thể** đạt được hàng rào bảo vệ hoàn hảo, in ra -1 trên dòng đầu tiên.

Nếu có thể đạt được hàng rào bảo vệ hoàn hảo, in ra **tổng quãng đường di chuyển tối thiểu** mà robot phải thực hiện để đạt được. Giá trị này luôn là một số nguyên.

Ràng buộc

- Subtask 1 (10% số điểm): $N \leq 10$, $L \leq 100$, $r \leq 10$
- Subtask 2 (10% số điểm): $N \leq 30$, $L \leq 2000$, $r \leq 30$

- Subtask 3 (15% số điểm): $N \leq 500$
- Subtask 4 (30% số điểm): $N \leq 2500$
- Subtask 5 (35% số điểm): Không có ràng buộc bổ sung.

Ví dụ

ROBOT.INP	ROBOT.OUT
5 20 2 4 10 11 17 19	36
5 18 2 7 10 12 14 18	22
4 18 3 8 10 14 18	17
1 33 4 33	-1