**NGÂN HÀNG CÂU HỎI THI KẾT THÚC HỌC PHẦN – THI PHÒNG MÁY**

**Môn:** Cấu trúc dữ liệu và giải thuật. **Phần:** Đồ thị

**MỤC LỤC**

[**BÀI 1. DSA\_P011. ĐẾM SỐ VẬT CẢN TRÊN MÊ CUNG** 2](#_Toc207001362)

[**BÀI 2. DSA\_P071. LIÊN THÔNG MẠNH** 2](#_Toc207001363)

[**BÀI 3. DSA\_P072. ĐƯỜNG ĐI BFS** 3](#_Toc207001364)

[**BÀI 4. DSA\_P073. CHU TRÌNH TRÊN ĐỒ THỊ CÓ HƯỚNG** 4](#_Toc207001365)

[**BÀI 5. DSA\_P074. THÀNH PHẦN LIÊN THÔNG – 1** 4](#_Toc207001366)

[**BÀI 6. DSA\_P075. ĐI HỌC** 5](#_Toc207001367)

[**BÀI 7. DSA\_P076. ĐỈNH TRỤ VÀ CẠNH CẦU** 6](#_Toc207001368)

[**BÀI 8. DSA\_P078. THÀNH PHẦN LIÊN THÔNG – 2** 7](#_Toc207001369)

[**BÀI 9. DSA\_P079. QUAY LẠI ĐỈNH 1** 8](#_Toc207001370)

[**BÀI 10. DSA\_P080. KHÔNG LIÊN THÔNG VỚI ĐỈNH 1** 9](#_Toc207001371)

[**BÀI 11. DSA\_P081. CHIA CẮT ĐỒ THỊ** 10](#_Toc207001372)

[**BÀI 12. DSA\_P082. ĐƯỜNG ĐI BFS** 10](#_Toc207001373)

[**BÀI 13. DSA\_P089. ĐỈNH THẮT** 11](#_Toc207001374)

[**BÀI 14. DSA\_P113. TỔ TIÊN CHUNG GẦN NHẤT** 11](#_Toc207001375)

[**BÀI 15. DSA\_P121. ĐIỂM NGHẼN GIAO THÔNG** 12](#_Toc207001376)

[**BÀI 16. DSA\_P122. ĐƯỜNG ĐI TRUNG BÌNH NGẮN NHẤT** 13](#_Toc207001377)

[**BÀI 17. DSA\_P123. DI CHUYỂN TRÊN ĐỒ THỊ CÓ TRỌNG SỐ** 13](#_Toc207001378)

[**BÀI 18. DSA\_P125. BỔ SUNG CẠNH** 14](#_Toc207001379)

[**BÀI 19. DSA\_P126. MÊ CUNG** 14](#_Toc207001380)

[**BÀI 20. DSA\_P127. ĐƯỜNG ĐI DFS – BFS + DSA\_P221. ĐƯỜNG ĐI DFS** 15](#_Toc207001381)

[**BÀI 21. DSA\_P133. DI CHUYỂN ROBOT + DSAKT074. ROBOCON** 16](#_Toc207001382)

[**BÀI 22. DSA\_P139. SỐ LẦN DUYỆT ÍT NHẤT** 18](#_Toc207001383)

[**BÀI 23. DSA\_P208. BÌNH THÔNG NHAU** 18](#_Toc207001384)

[**BÀI 24. DSA\_P209. BÀI TOÁN SO SÁNH** 19](#_Toc207001385)

[**BÀI 25. DSA\_P302. HỌC NHÓM** 20](#_Toc207001386)

[**BÀI 26. DSA\_P305. XOÁ CẠNH** 21](#_Toc207001387)

[**BÀI 27. DSAKT077. TRÒ CHƠI GHÉP CHỮ** 22](#_Toc207001388)

[**BÀI 28. DSAKT121. DUYỆT ĐỒ THỊ** 23](#_Toc207001389)

[**BÀI 29. DSAKT122. ĐỒ THỊ LIÊN THÔNG – 1** 24](#_Toc207001390)

[**BÀI 30. DSAKT123. ĐỒ THỊ LIÊN THÔNG – 2** 25](#_Toc207001391)

[**BÀI 31. HÌNH CHỮ NHẬT NHỎ NHẤT** 25](#_Toc207001392)

[**BÀI 32. CÂY KHUNG NHỎ NHẤT** 26](#_Toc207001393)

[**BÀI 33. KIỂM TRA ĐỒ THỊ Ơ-LE** 27](#_Toc207001394)

[**BÀI 33. ĐƯỜNG KÍNH CỦA CÂY** 27](#_Toc207001395)

[**BÀI 34. DSATest28. HỆ THỐNG GIAO THÔNG** 28](#_Toc207001396)

# **BÀI 1. DSA\_P011. ĐẾM SỐ VẬT CẢN TRÊN MÊ CUNG**

Một mê cung được mô tả dưới dạng ma trận ký tự trong đó dấu ‘.’ là mô tả ô trống, không có vật cản, dấu ‘#’ mô tả một vật cản. Các vật cản sẽ ghép lại với nhau thành vật cản lớn hơn nếu nó liền kề theo hàng hoặc cột.

Hãy đếm xem trong mê cung có bao nhiêu vật cản.

**Input**

Dong đầu ghi số hai số N, M (N, M ≤ 1000) là số hàng và số cột của mê cung.

N dòng tiếp theo mô tả mê cung trong đó chỉ có các ký tự ‘.’ và ‘#’.

**Output**. Ghi ra số vật cản đếm được.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 6  .#....  ..#...  ..#..#  ...##.  .#.... | 5 |

**Phân dạng:** Đếm số TPLT

# **BÀI 2. DSA\_P071. LIÊN THÔNG MẠNH**

Cho đồ thị có hướng có N đỉnh và M cạnh. Đếm số thành phần liên thông mạnh của đồ thị.

**Input.**

* Dòng đầu tiên là hai số N và M (1 ≤ N ≤ 104, 1 ≤ M ≤ 105)
* M dòng tiếp theo, mỗi dòng gồm 2 số u và v, mô tả đường 1 chiều từ u đến v

**Output.**

* Một số nguyên duy nhất là số thành phần liên thông mạnh tìm được

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 2  1 2  2 3 | 3 |
| 4 4  1 2  2 3  3 4  4 1 | 1 |

**Phân dạng:** Kosaraju

# **BÀI 3. DSA\_P072. ĐƯỜNG ĐI BFS**

Cho đồ thị có hướng G = <V, E> được biểu diễn dưới dạng danh sách cạnh.

Hãy tìm đường đi từ đỉnh u đến đỉnh v trên đồ thị bằng thuật toán BFS.

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm |E|+1 dòng:

+) Dòng đầu tiên đưa vào bốn số |V|, |E|, u, v tương ứng với số đỉnh, số cạnh, đỉnh xuất phát u, đỉnh kết thúc v;

+) |E| dòng tiếp theo mỗi dòng đưa vào bộ đôi x, y tương ứng với một cạnh của đồ thị.

* T, |V|, |E| thỏa mãn ràng buộc: 1 ≤ T ≤ 100; 1 ≤ |V| ≤ 103 ; 1 ≤ |E| ≤ |V| (|V| - 1) / 2;

**Output:**

Đưa ra đường đi từ đỉnh s đến đỉnh t của mỗi test theo thuật toán BFS của mỗi test theo khuôn dạng của ví dụ dưới đây. Nếu không có đáp án, in ra -1.

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  6 9 1 6  1 2  2 5  3 1  3 2  3 5  4 3  5 4  5 6  6 4 | 1 2 5 6 |

# **BÀI 4. DSA\_P073. CHU TRÌNH TRÊN ĐỒ THỊ CÓ HƯỚNG**

Cho đồ thị có hướng G = <V, E> được biểu diễn dưới dạng danh sách cạnh.

Kiểm tra xem đồ thị có tồn tại chu trình hay không?

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm |E| + 1 dòng:

Dòng đầu tiên đưa vào hai số |V|, |E| tương ứng với số đỉnh, số cạnh của đồ thị;

E dòng tiếp theo, mỗi dòng gồm 2 số x và y là điểm đầu và điểm cuối của cạnh đồ thị

* T, |V|, |E| thỏa mãn ràng buộc: 1≤T≤100; 1≤|V|≤103; 1≤|E|≤|V|(|V|-1)/2;

**Output:**

* Đưa ra YES hoặc “NO” kết quả test theo từng dòng tương ứng với đồ thị tồn tại hoặc không tồn tại chu trình.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  6 9  1 2  2 4  3 1  3 2  3 5  4 3  5 4  5 6  6 4 | YES |

# **BÀI 5. DSA\_P074. THÀNH PHẦN LIÊN THÔNG – 1**

Cho đồ thị vô hướng G có N đỉnh, M cạnh. Hãy liệt kê các đỉnh không cùng thành phần liên thông với một đỉnh cho trước.

**Input.**

Dòng đầu ghi 3 số N, M và X (0 < N < 300; 1 ≤ M ≤ N \* (N - 1) / 2), 0 < X < N).

Tiếp theo là M dòng, mỗi dòng ghi một cạnh của đồ thị.

Các cạnh được liệt kê với thứ tự bất kỳ.

**Output.**

Ghi ra các đỉnh không liên thông với đỉnh X theo thứ tự tăng dần, mỗi dòng ghi một đỉnh. Nếu không có đỉnh nào thì ghi ra số 0.

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 6 4 2  1 3  2 3  1 2  4 5 | 4  5  6 |

# **BÀI 6. DSA\_P075. ĐI HỌC**

Tí và Tèo là bạn học cùng nhau hồi cấp 1. Lên cấp 2, Tí chuyển nhà nên phải học ở trường của xã bên cạnh. Tuy nhiên, 2 bạn vẫn rất thân với nhau và hay thường đợi nhau cùng đi học, cho dù chỉ đi chung 1 quãng đường hoặc gặp mặt nhau một cái.

Cho biết địa điểm nhà Tí là H1, trường của Tí là S1, nhà của Tèo là H2, trường của Tèo là S2 (4 địa điểm này khác nhau). Có M tuyến đường hai chiều biểu diễn mô hình giao thông tại địa phương của hai bạn. Biết rằng Tí và Tèo luôn đi theo con đường ngắn nhất để đến trường, và tốc độ của 2 bạn bằng nhau (1 đơn vị độ dài / 1 đơn vị thời gian). Hai bạn xuất phát cùng một lúc và phải di chuyển liên tục, trừ trường hợp đã tới trường học của mình.

Các bạn hãy xác định xem Tí và Tèo có thể gặp nhau tại bao nhiêu địa điểm? Trường hợp Tèo đã đến trường, mà trường học của Tèo lại nằm trên đường đi học của Tí, thì bạn ấy sẽ đứng đợi ở trường cho tới khi gặp được Tí đi học qua (và ngược lại).

**Input.**

* Dòng đầu tiên là số lượng bộ test T (T ≤ 20).
* Mỗi test bắt đầu bởi hai số nguyên N, M (N, M ≤ 500 000).
* Tiếp theo là 4 số nguyên H1, S1, H2, S2.
* M dòng tiếp theo, mỗi dòng gồm 3 số nguyên u, v, c cho biết có cạnh nối giữa u và v, độ dài bằng c (u != v, c ≤ 109). Input đảm bảo đồ thị đã liên thông.

**Output.** Với mỗi test, in ra số lượng địa điểm có thể là nơi hai bạn sẽ gặp nhau trên đường đi học. Nếu đáp số là vô hạn, in ra “infinity”

**Ví dụ.**

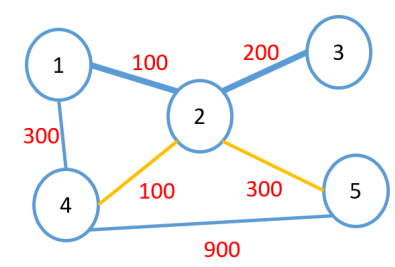
|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  5 6  1 3 4 5  1 2 100  2 3 200  1 4 300  4 2 100  2 5 300  4 5 900  3 2  1 3 2 3  1 3 100  2 3 200  5 6  1 3 4 5  1 2 100  2 3 200  1 3 200  4 2 100  2 5 300  4 5 400  6 6  1 5 2 6  1 3 100  2 3 100  3 4 200  4 5 100  4 6 300 | 1  1  0  infinity |

**Giải thích test.**

Giải thích test 1: Điểm chung duy nhất là đỉnh 2.

Giải thích test 2: Đây là trường hợp đứng đợi tại trường học của bạn thứ nhất.

Giải thích test 4: 2 bạn đi chung trên con đường 3 trùng 4, và có vô số điểm chung nhau.

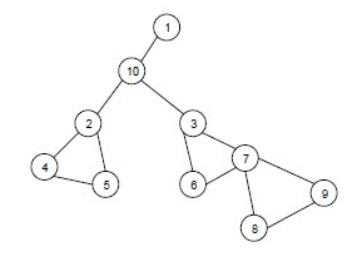
****

# **BÀI 7. DSA\_P076. ĐỈNH TRỤ VÀ CẠNH CẦU**

Cho đồ thị vô hướng có N đỉnh (1 ≤ N ≤ 104) và M cạnh (1 ≤ M ≤ 105). Ta nhớ lại các định nghĩa sau:

* Một đỉnh được gọi là đỉnh trụ nếu như xóa đỉnh đó đi sẽ làm tăng số lượng thành phần liên thông của đồ thị.
* Một cạnh được gọi là cạnh cầu nếu như xóa cạnh đó đi sẽ làm tăng số lượng thành phần liên thông của đồ thị.

Nhiệm vụ của bạn là hãy đếm số lượng đỉnh trụ và cạnh cầu của đồ thị đã cho.



**Input:**

* Dòng đầu tiên hai số nguyên N và M.
* M dòng tiếp theo, mỗi dòng gồm hai số nguyên u, v cho biết có cạnh nối giữa đỉnh u và v.

**Output.** In ra 2 số nguyên là số lượng đỉnh trụ và cạnh cầu của đồ thị

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 10 12  1 10  10 2  10 3  2 4  4 5  5 2  3 6  6 7  7 3  7 8  8 9  9 7 | 4 3 |

# **BÀI 8. DSA\_P078. THÀNH PHẦN LIÊN THÔNG – 2**

Cho đồ thị vô hướng gồm N đỉnh và M cạnh. Xét lần lượt các đỉnh từ 1 đến N, hãy tính xem nếu xóa đỉnh đó đi thì đồ thị có bao nhiêu thành phần liên thông.

**Input.**

* Dòng đầu ghi 2 số N và M (1 < N < 20000; 1 < M < 50000).
* Tiếp theo là M dòng ghi các cạnh của đồ thị.

**Output.**

In ra N dòng, dòng thứ k (tính từ 1) là số thành phần liên thông của đồ thị nếu xóa đỉnh K

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4 3  1 2  2 3  2 4 | 1  3  1  1 |

# **BÀI 9. DSA\_P079. QUAY LẠI ĐỈNH 1**

Cho đồ thị **có hướng**với N đỉnh và M cạnh. Người ta muốn thực hiện hành trình với hai bước di chuyển sau:

- Bước 1: Tìm đường đi từ đỉnh 1 qua các cạnh đến đỉnh 2.

- Bước 2: Từ đỉnh 2 lại đi qua các cạnh nào đó để quay lại đỉnh 1.

Trên mỗi chặng đường (1 → 2 hoặc 2 → 1) thì không được đi qua một cạnh nào 2 lần. Hãy tính xem số đỉnh ít nhất cần phải đi qua trong cả hành trình đó là bao nhiêu.

**Input**

* Dòng đầu ghi số bộ test.
* Mỗi test bắt đầu với một dòng ghi hai số N, M (1 < N ≤ 20).
* Tiếp theo là M dòng ghi các cạnh có hướng. Không có cạnh nào trùng nhau.

**Output**

Với mỗi bộ test, ghi ra số đỉnh tối thiểu cần phải đi qua thỏa mãn yêu cầu đề bài.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Ouput** |
| 2  6 7  1 3  3 4  4 5  5 1  4 2  2 6  6 3  9 11  1 3  3 4  4 2  2 5  5 3  3 6  6 1  2 7  7 8  8 9  9 1 | 6  6 |

**Giải thích test 1:**

|  |  |
| --- | --- |
|  | * Đường đi ngắn nhất từ 1 🡪 2 (Gọi là Path 1) là: 1 🡪 3 🡪 4 🡪 2 * Đường đi ngắn nhất từ 2 🡪 1 (Gọi là Path 2) là: 2 🡪 6 🡪 3 🡪 4 🡪 5 🡪 1 * Tập các đỉnh phân biệt trong cả 2 đường là {1, 2, 3, 4, 5, 6}. Do vậy, đáp số là 6 đỉnh * Cần lưu ý rằng: Chỉ khi nào 1 cạnh có mặt trong 1 path 2 lần thì mới phạm luật. Trong ví dụ này, cạnh 3 🡪 4 xuất hiện 2 lần, nhưng là ở 2 path khác nhau. Xét riêng mỗi path thì cạnh này xuất hiện mỗi path 1 lần nên cạnh này không được xem là cạnh lặp |

**Giới hạn thời gian:** 1s

**Giới hạn bộ nhớ:** 200000 Kb

# **BÀI 10. DSA\_P080. KHÔNG LIÊN THÔNG VỚI ĐỈNH 1**

Cho đồ thị vô hướng G có N đỉnh, M cạnh. Hãy liệt kê các đỉnh không cùng thành phần liên thông với đỉnh 1.

**Input.**

* Dòng đầu ghi 2 số N và M (0 < N < 300; 1 ≤ M ≤ N \* (N - 1) / 2)).
* Tiếp theo là M dòng, mỗi dòng ghi một cạnh của đồ thị. Các cạnh được liệt kê với thứ tự bất kỳ.

**Output.**

Ghi ra các đỉnh không liên thông với đỉnh 1 theo thứ tự tăng dần, mỗi dòng ghi một đỉnh. Nếu không có đỉnh nào thì ghi ra số 0.  
**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 6 4  1 3  2 3  1 2  4 5 | 4  5  6 |

# **BÀI 11. DSA\_P081. CHIA CẮT ĐỒ THỊ**

Cho đồ thị vô hướng G có N đỉnh và M cạnh. Hãy tìm đỉnh u sao cho nếu loại bỏ đỉnh u ra khỏi đồ thị thì đồ thị bị chia cắt thành nhiều thành phần liên thông nhất.

**Input.**

Dòng đầu ghi số bộ test, mỗi bộ test gồm:

* Dòng đầu ghi số N là số đỉnh (1 < N < 100) và số M là số cạnh (M < N\*(N-1)/2).
* M dòng tiếp theo ghi các cạnh của đồ thị.

**Output.**

* Ghi ra thứ tự đỉnh (tính từ 1) thỏa mãn nếu loại bỏ đỉnh đó ra khỏi đồ thị thì sẽ chia cắt ra nhiều thành phần liên thông nhất.
* Nếu có nhiều hơn 1 đỉnh thỏa mãn thì in ra thứ tự đỉnh nhỏ nhất.
* Nếu không thể chia cắt được đồ thị thì ghi ra 0.

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  5 5  1 2  1 3  2 3  3 4  3 5  5 7  1 2  1 3  2 3  2 5  3 4  3 5  4 5 | 3  0 |

# **BÀI 12. DSA\_P082. ĐƯỜNG ĐI BFS**

Cho đồ thị vô hướng G = <V, E> được biểu diễn dưới dạng danh sách cạnh.

Hãy tìm đường đi từ đỉnh s đến tất cả các đỉnh còn lại bằng thuật toán BFS.

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng:

+) Dòng đầu tiên đưa vào ba số N, M, s tương ứng với số đỉnh, số cạnh, đỉnh xuất phát. +) Dòng tiếp theo đưa vào các bộ đôi u, v tương ứng với một cạnh của đồ thị.

* T, N, M thỏa mãn ràng buộc: 1 ≤ T ≤ 100; 1 ≤ N ≤ 103 ; 1≤ M ≤ N \* (N - 1) / 2;

**Output**

* Đưa ra đường đi từ đỉnh s đến lần lượt từng đỉnh từ 1 đến N (trừ chính đỉnh s).
* Nếu đỉnh nào không có đường đi từ s thì ghi ra “No path”

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  6 9 1  1 2 1 3 2 3 2 5 3 4 3 5 4 5 4 6 5 6 | 1 2  1 3  1 3 4  1 2 5  1 2 5 6 |

# **BÀI 13. DSA\_P089. ĐỈNH THẮT**

Cho đồ thị có hướng liên thông G có N đỉnh và M cạnh. Với một cặp đỉnh (u,v), đỉnh thắt của cặp đỉnh này được định nghĩa là một đỉnh mà tất cả đường đi từ u tới v đều đi qua nó. Hãy đếm số đỉnh thắt với cặp đỉnh (u,v).

**Input.**

* Dòng đầu ghi số bộ test, không quá 100.
* Mỗi bộ test bắt đầu với một dòng ghi 4 số N, M, u, v

(0 < N ≤ 100; 1 < M ≤1000; 1 ≤ u,v ≤ N).

* Tiếp theo là M dòng ghi các cạnh của đồ thị

**Output.** Với mỗi bộ test, ghi ra số đỉnh thắt của cặp đỉnh (u,v)

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  5 7 1 3  1 2  2 4  2 5  3 1  3 2  4 3  5 4  4 5 1 4  1 2  1 3  2 3  2 4  3 4 | 2  0 |

# **BÀI 14. DSA\_P113. TỔ TIÊN CHUNG GẦN NHẤT**

Cho một cây có N đỉnh có gốc tại đỉnh 1. Có Q truy vấn, mỗi truy vấn yêu cầu bạn tìm đỉnh cha chung gần nhất giữa 2 đỉnh u và v.

**Input:**

* Dòng đầu tiên chứa số nguyên N (2 ≤ N ≤ 105).
* N - 1 dòng tiếp theo, mỗi dòng chứa 2 số nguyên U và V cho biết có cạnh giữa đỉnh U và V.
* Dòng tiếp theo chứa số nguyên M là số lượng truy vấn (1 ≤ M ≤ 105).
* M dòng tiếp theo, mỗi dòng gồm 2 số nguyên X và Y.

**Output:**

Với mỗi truy vấn, in ra đáp án là cha chung gần nhất của X và Y trên một dòng

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 6  1 2  1 3  2 4  2 5  4 6  3  4 6  5 6  3 6 | 4  2  1 |

# **BÀI 15. DSA\_P121. ĐIỂM NGHẼN GIAO THÔNG**

Cho hệ thống giao thông gồm N điểm. Biết giữa hai điểm bất kỳ của hệ thống đều tồn tại đường đi trực tiếp hoặc gián tiếp thông qua một số điểm trung gian. Ta gọi điểm giao thông s là điểm nghẽn giao thông của cặp nút giao thông u, v nếu mọi đường đi từ u đến v đều phải đi qua s.

Nhiệm vụ của bạn là viết một chương trình tìm số lượng các đỉnh thắt s của cặp điểm u, v của hệ thống giao thông.

**Input**

* Dòng đầu tiên chứa số nguyên dương không lớn hơn 100 là số lượng các bộ dữ liệu T.
* Mỗi bộ dữ liệu gồm một nhóm dòng theo khuôn dạng:
  + Dòng 1 chứa 4 số nguyên N, M, u, v (u, v, N ≤ 100; M ≤ 1000).
  + M dòng sau, mỗi dòng ghi hai số i, j cách nhau một khoảng trống cho biết có đường nối trực tiếp giữa i với j (1 ≤ i, j ≤ N).

**Output.**

Với mỗi bộ dữ liệu, đưa ra màn hình một số nguyên là số lượng điểm nghẽn giao thông của cặp điểm u,v tương ứng.

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  5 7 1 3  1 2  2 4  2 5  3 1  3 2  4 3  5 4  4 5 1 4  1 2  1 3  2 3  2 4  3 4 | 2  0 |

# **BÀI 16. DSA\_P122. ĐƯỜNG ĐI TRUNG BÌNH NGẮN NHẤT**

Tại đất nước Highland có N thành phố, mỗi cặp thành phố được kết nối với nhau bởi một tuyển đường một chiều duy nhất. Chi phi di chuyển giữa thành phố thủ u tới thành phố v là C[u][v].

Bạn cần tìm một hành trình thỏa mãn các yêu cầu:

* Có thể xuất phát ở bất cứ đâu, nhưng kết thúc hành trình phải quay lại điểm xuất phát.
* Phải đi qua ít nhất 2 tuyến đường.
* Chi phi trung bình trên mỗi tuyển đường là nhỏ nhất (Lấy tổng chi phi / số tuyến đường đi qua).

**Input**

* Dòng đầu tiên là số lượng thành phố N(2 ≤ N ≤ 500).
* N dòng tiếp theo, mỗi dòng gồm N số nguyên cho biết chi phi di chuyển từ thành phố u tới v. C[u][u] = 0 và 1 ≤ C[u][v] ≤ 200.

**Output:**

In ra chi phí trung binh trên mỗi tuyến đường nhỏ nhất tìm được, dưới dạng phân số , trong đó ước chung lớn nhất của A và B bằng 1.

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  0 1  2 0 | 3/2 |
| 3  0 2 6  4 0 2  1 9 0 | 5/3 |

# **BÀI 17. DSA\_P123. DI CHUYỂN TRÊN ĐỒ THỊ CÓ TRỌNG SỐ**

Cho đồ thị có trọng số với N đỉnh và M cạnh có hướng. Với mỗi cặp đinh (S,T), hãy tính giá trị trọng số X, là trọng số lớn nhất trong số các cạnh phải đi qua để di chuyển từ S đến T. Biết rằng quy tắc di chuyển là luôn mong muốn giá trị X là nhỏ nhất có thể.

**Input**

* + Dòng đầu ghi 3 số N, M và Q
* N là số đỉnh của đồ thị (1 ≤ N ≤ 300)
* M là số lượng cạnh có hướng (1≤ M ≤ 25000)
* Q là số truy vấn

- Tiếp theo là M dòng, mỗi dòng ghi 3 số u,v,w là một cạnh của đồ thị. Trọng số w đảm bảo nguyên dương và không quá 106.

- Tiếp theo là Q dòng, mỗi dòng ghi một cặp đỉnh S và T.

**Output**

Với mỗi truy vấn, hãy in ra giá trị trọng số X tìm được. Nếu không có đường đi từ S đến T thì in ra -1

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |

# **BÀI 18. DSA\_P125. BỔ SUNG CẠNH**

Cho đồ thị vô hướng G có thể chưa liên thông. Từ đỉnh 1, bằng cách duyệt đồ thị ta sẽ đánh dấu được các đỉnh cùng thành phần liên thông với đỉnh 1.

Người ta muốn đánh dấu được nhiều đỉnh hơn bằng cách bổ sung một cạnh duy nhất vào đồ thị. Hãy đếm số đỉnh nhiều nhất có thể đánh dấu theo mô tả trên. **[DSU]**

**Input.**

* Dòng đầu ghi 2 số N và M là số đỉnh và số cạnh của đồ thị. (1 ≤ N,M ≤ 105).
* M dòng tiếp theo, mỗi dòng ghi một cạnh của đồ thị.

**Output.**

Đưa ra số lượng đỉnh lớn nhất có thể được đánh dấu bằng cách đi từ đỉnh 1 nếu bổ sung một cạnh vào đồ thị.

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 2  1 2  3 2 | 3 |
| 5 3  1 4  4 2  2 1 | 4 |

# **BÀI 19. DSA\_P126. MÊ CUNG**

Mê cung có N cửa vào đánh số từ 1 đến N. Với thiết kế mới, người chơi xuất phát từ cửa này thì đi lòng vòng 1 hồi sẽ đến một cửa khác, sau đó có thể từ đó đi tiếp. Một người chơi thắng cuộc nếu chọn được một cửa xuất phát nào đó mà sau một số lần di chuyển (không đi lại con đường cũ) sẽ quay lại được đúng cửa ban đầu.

Bạn là người thiết kế mê cung nên bạn cần kiểm tra tính khả thi của trò chơi. Tức là bạn sẽ phải thiết kế trước các tuyến đường giữa các cửa của mê cung sao cho phải có ít nhất một cách di chuyển nào đó giúp người chơi thắng cuộc. Cho dù thực tế việc di chuyển trong mê cung không hề dễ dàng.

Cho trước danh sách các tuyến đường. Hãy kiểm tra xem liệu có thể có cách nào giúp người chơi thắng cuộc được hay không.

**Input**

- Dòng đầu ghi số bộ test.

- Với mỗi bộ test:

• Dòng đầu ghi 2 số N và M (1 ≤ N, M ≤ 105).

• M dòng tiếp theo, mỗi dòng ghi một tuyến đường trong thiết kế của mê cung.

**Output.**

Với mỗi bộ test, ghi ra YES nếu có tuyến đường giúp người chơi thắng cuộc. Ghi ra NO nếu ngược lại

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  4 3  1 2  1 4  3 4  4 4  1 2  1 3  1 4  2 4 | NO  YES |

# **BÀI 20. DSA\_P127. ĐƯỜNG ĐI DFS – BFS + DSA\_P221. ĐƯỜNG ĐI DFS**

Cho đồ thị vô hướng G có N đỉnh và M cạnh. Hãy tìm và in ra:

* Đường đi từ đỉnh 1 đến các đỉnh khác bằng thuật toán DFS
* Đường đi ngược lại từ các đỉnh khác về 1 bằng thuật toán BFS.

**Input.**

* Dòng đầu ghi 2 số N và M (1 ≤ N ≤ 1000; 1 ≤ M ≤ 2000).
* M dòng tiếp theo, mỗi dòng ghi một cạnh vô hướng của đồ thị.

**Output.** Xét từng đỉnh từ 2 đến N, với mỗi đỉnh ghi ra 2 dòng:

* Dòng đầu ghi đường đi từ 1 đến đỉnh đó theo DFS
* Dòng tiếp theo ghi đường đi từ đỉnh đó về 1 theo thuật toán BFS

Nếu không có đường đi thì ghi ra -1

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4 4  1 2  1 3  1 4  3 4 | 1 2  2 1  1 3  3 1  1 3 4  4 1 |

# **BÀI 21. DSA\_P133. DI CHUYỂN ROBOT + DSAKT074. ROBOCON**

Luật chơi của cuộc thi Robocon năm 2022 được quy định như sau:

* + Các robot sẽ thi đấu trên một bản đồ gồm:
* N ô (đỉnh) vị trí khác nhau được đánh số từ 1 đến N
* M đoạn đường một chiều nối các cặp đỉnh.
  + Mỗi ô được tô bằng một trong ba màu 1, 2, 3.
  + Mỗi đoạn đường cũng được tô bằng một trong ba màu 1, 2, 3 nhưng sẽ bị biến đổi màu theo thời gian. Cụ thể, nếu ở thời điểm t, đoạn đường có màu là c thì ở thời điểm t + 1 đoạn đường sẽ có màu là (c mod 3)+1.
  + Mỗi đội chơi phải điều khiển hai robot cùng di chuyển trên bản đồ để hai robot gặp nhau sớm nhất tại một ô.
  + Ban đầu (tại thời điểm 0), robot thứ nhất đứng ở ô 1, robot thứ hai đứng ở ô 2, quy tắc di chuyển của hai robot là:
* Mỗi đơn vị thời gian, cả hai robot cùng phải di chuyển qua một đoạn đường;
* Giả sử ở thời điểm t robot 1 đang đứng ở ô i, robot 2 đang đứng ở ô j, khi đó:

+) Robot 1 buộc phải di chuyển theo một trong các đoạn đường có màu ở thời điểm t cùng màu với màu ở ô j

+) Robot 2 buộc phải di chuyển theo một trong các đoạn đường có màu ở thời điểm t cùng màu với màu ở ô i.

**Yêu cầu:** Cho thông tin về bản đồ với màu ban đầu của các đoạn đường, hãy xác định thời điểm hai robot gặp nhau sớm nhất.

**Input**

* Dòng đầu tiên là số lượng bộ test T (T ≤ 10).
* Mỗi test bắt đầu bằng 2 số nguyên N và M (M ≤ ).
* Dòng thứ hai ghi N số nguyên, là màu của ô thứ i.
* M dòng tiếp, mỗi dòng gồm 3 số nguyên u, v, c mô tả đoạn đường từ u đến v có màu tại thời điểm 0 là c.

**Giới hạn:**

* **Subtask 1:** 2 ≤ N ≤ 10
* **Subtask 2:** 2 ≤ N ≤ 50
* **Subtask 3:** 2 ≤ N ≤ 200

**Output**. Với mỗi test, hãy in ra đáp án trên một dòng. Nếu không tìm được đáp án, in ra -1.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  3 4  1 2 3  1 2 2  2 1 2  2 3 1  3 1 1  3 4  1 2 3  1 2 3  2 1 2  2 3 1  3 1 1 | 2  -1 |

**Giải thích test**

* **Test 1:**

1. **Ban đầu (t = 0):**
   * Robot 1 đang ở đỉnh 1 (màu 1).
   * Robot 2 đang ở đỉnh 2 (màu 2).

**Theo luật:**

* + Robot 1 phải đi theo một cạnh xuất phát từ đỉnh 1 có màu bằng với màu đỉnh 2 (Tức màu 2). Mà Cạnh 1 → 2 có màu 2 ⇒ Robot 1 có thể đi từ 1 đến 2.
  + Robot 2 phải đi theo một cạnh xuất phát từ đỉnh 2 có màu bằng với màu đỉnh 1 (Tức màu 1). Mà Cạnh 2 → 3 có màu 1 ⇒ Robot 2 đi từ 2 đến 3.

**Sau khi di chuyển:**

* + Robot 1 ở đỉnh 2.
  + Robot 2 ở đỉnh 3.

1. **Thời điểm t = 1:**

Bây giờ các cạnh đã đổi màu (Quy luật: Màu c chuyển thành (c mod 3) + 1).

* + Cạnh 2→1: Lúc đầu màu 2, sang t = 1 trở thành màu 3.
  + Cạnh 3→1: Lúc đầu màu 1, sang t = 1 trở thành màu 2.

**Vị trí robot:**

* + Robot 1 ở đỉnh 2 (màu 2).
  + Robot 2 ở đỉnh 3 (màu 3)

**Theo luật:**

* + Robot 1 phải chọn cạnh xuất phát từ 2 có màu bằng màu đỉnh 3 (màu 3). → Có cạnh 2 → 1 (Màu 3 ở thời điểm này).
  + Robot 2 phải chọn cạnh xuất phát từ 3 có màu bằng màu đỉnh 2 (màu 2). → Có cạnh 3 → 1 (Màu 2 ở thời điểm này).

**Sau khi di chuyển:**

* + Robot 1 từ 2 → 1.
  + Robot 2 từ 3 → 1.

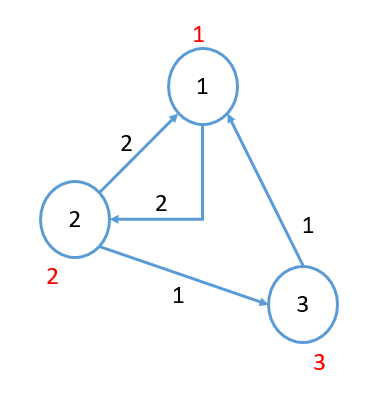
1. **Thời điểm t = 2:**

Cả hai robot đều đang đứng ở đỉnh 1 ⇒ hai robot gặp nhau.

Vậy đáp án là 2.

* **Test 2:**

Tại t = 0, đỉnh 1 không có cạnh nào có màu bằng 2, do đó cần in ra -1.



# **BÀI 22. DSA\_P139. SỐ LẦN DUYỆT ÍT NHẤT**

Cho đồ thị có hướng với N đỉnh và M cạnh. Mỗi lần duyệt đồ thị, bạn có thể lựa chọn bắt đầu từ bất cứ đỉnh nảo và cố gắng đi qua nhiều đỉnh nhất có thể (đỉnh nào đã đi qua sẽ được đánh dấu). Hãy tính xem cần ít nhất bao nhiêu lần duyệt để đi qua tất cả các đỉnh của đồ thị.

**Input.**

* Dòng đầu ghi 2 số N và M (1 ≤ N ≤ 1000; 0 ≤ M ≤ 10000)
* M dòng sau ghi các cạnh có hướng của đồ thị.

**Output.**

Ghi ra một số nguyên duy nhất là số lần duyệt ít nhất tính được.

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 4  1 2  1 3  4 1  5 1 | 2 |

# **BÀI 23. DSA\_P208. BÌNH THÔNG NHAU**

Có N thùng nước được đánh số từ 1 đến N, giữa 2 thùng bất kỳ đều có một ống nối có một van có thể khóa hoặc mở. Ở trạng thái ban đầu tất cả các van đều đóng.

Bạn được cho một số yêu cầu, trong đó mỗi yêu cầu có 2 dạng:

* Dạng X Y 1 có ý nghĩa là bạn cần mở van nối giữa 2 thùng X và Y.
* Dạng X Y 2 có ý nghĩa là bạn cần cho biết với trạng thái các van đang mở / khóa như hiện tại thì 2 thùng X và Y có thuộc cùng một nhóm bình thông nhau hay không?

Hai thùng được coi là thuộc cùng một nhóm bình thông nhau nếu nước từ bình nàycó thể chảy đến được bình kia qua một số ống có van đang mở. **[DSU]**

**Input.**

Dòng đầu tiên là số lượng truy vấn Q (Q ≤ 105).

Mỗi truy vấn gồm 3 số nguyên X, Y, Z (X, Y ≤ 105).

**Output.** Với mỗi truy vấn, in ra đáp án tìm được trên một dòng.

**Ví du.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 9  1 2 2  1 2 1  3 7 2  2 3 1  1 3 2  2 4 2  1 4 1  3 4 2  1 7 2 | 0  0  1  0  1  0 |

# **BÀI 24. DSA\_P209. BÀI TOÁN SO SÁNH**

Cho một dãy các phép so sánh chiều cao giữa các sinh viên, mỗi sinh viên được đại diện bởi tên sinh viên (Dãy ký tự không có khoảng trống và không có 2 sinh viên nào trùng tên). Hãy chỉ ra liệu có thể tất cả các phép so sánh đó đều đúng hay không?

**[Kahn, chu trình trên đồ thị có hướng]**

**Input**.

* Dòng đầu ghi số N là số phép so sánh. (1 ≤ N ≤ 105).
* Mỗi phép so sánh gồm 2 xâu ký tự đại diện cho 2 cái tên sinh viên, và một dấu lớn hơn hoặc nhỏ hơn, hai xâu ký tự đều không quá 20 chữ cái, không có khoảng trống.

**Output**

Ghi ra “possible” nếu tất cả phép so sánh đều có thể đúng hoặc “impossible” nếu ngược lại.

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  An > Binh  Binh > Cong  An < Cong | impossible |
| 3  An > Binh  Binh > Cong  An > Cong | possible |

# **BÀI 25. DSA\_P302. HỌC NHÓM**

Học kỳ 2 – Năm học 2023 – 2024, sinh viên khóa D22 (năm thứ 2) thuộc các ngành Kỹ thuật gồm: Công nghệ thông tin, An toàn thông tin, Khoa học máy tính, Công nghệ Đa phương tiện (tương ứng với D22CN, D22AT, D22KH, D22PT), cùng với khóa D21 (năm thứ 3) ngành Công nghệ Kỹ thuật Điện – Điện tử (D21DT) của Học viện Công nghệ Bưu chính Viễn thông – sau đây gọi chung là **D22** – đang chuẩn bị bước vào kỳ thi Kết thúc học phần môn **Cấu trúc dữ liệu và Giải thuật**. Đây là một môn học 3 tín chỉ, vốn nổi tiếng với tỉ lệ trượt luôn ở mức cao qua nhiều năm. Không chỉ riêng D22, mà còn có cả một số anh chị khóa D20, D19,… vẫn còn “lưu ban” ở trường và đang phải tiếp tục “điên đầu” để trả nợ môn học này. Trong bối cảnh học phí học lại ngày càng cao, tất cả sinh viên đều có cùng mục tiêu: phải qua môn ngay trong lần học đầu tiên, đồng thời nếu có thể thì sẽ phấn đấu để giành điểm cao. Chính vì vậy, D22 nhận thấy cần phải chăm chỉ hơn, đồng thời tìm ra những phương pháp học tập hiệu quả hơn, và một trong số những cách thức được lựa chọn chính là **kết nối lại với nhau**, lập nên các nhóm học tập, vừa để hỗ trợ lẫn nhau, vừa để tăng cơ hội giành được điểm A+.

Với quan điểm: Bạn của bạn cũng là bạn, mỗi người đưa ra thông tin về những người bạn của mình theo từng cặp. Trong cộng đồng sinh viên tất nhiên có những bạn chẳng chơi với ai khác ngoài chính mình. Ngoài ra, do khai báo thoải mái nên rất có thể nhiều cặp bạn bè sẽ bị lặp lại.

Hãy tính xem cuối cùng thì D22 có thể tạo nên mấy nhóm bạn bè cùng học chung với nhau môn Cấu trúc dữ liệu và Giải thuật nhé.

**Input**

* Dòng đầu ghi 2 số N và M lần lượt là số sinh viên và số cặp bạn bè được khai báo
* M dòng tiếp theo, mỗi dòng ghi một cặp bạn bè.
* Theo thống kê của Phòng Giáo vụ về đăng ký học phần, học kỳ này có khoảng 100.000 sinh viên các khoá (Học mới, học lại) đăng ký môn này
* Số cặp bạn bè không quá 200.000

**Output**. Ghi ra số nhóm bạn bè tính được.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 8 5  1 2  2 3  3 4  5 6  7 8 | 3 |

# **BÀI 26. DSA\_P305. XOÁ CẠNH**

Cho đồ thị vô hướng có N đỉnh và M cạnh. Biết rằng các đỉnh và các cạnh được đánh số từ 1. **Yêu cầu:** Cho q truy vấn, mỗi truy vấn gồm 2 số x và y. Bạn cần kiểm tra xem sau khi xoá cạnh thứ x thì còn bao nhiêu đỉnh cùng thành phần liên thông với đỉnh y

**Lưu ý:**

* Tất cả các thao tác đều chỉ được thực hiện lần lượt trên cùng đồ thị ban đầu.
* Sau khi xoá cạnh xong thì đồ thị không được khôi phục như ban đầu nữa.
* Trường hợp thao tác xoá bị trùng lặp, tức là cạnh ấy đã xoá trước đó rồi nhưng bây giờ lại chọn nó để xoá, thì coi như không có thao tác đó, vẫn trả lời truy vấn bình thường

**Input**

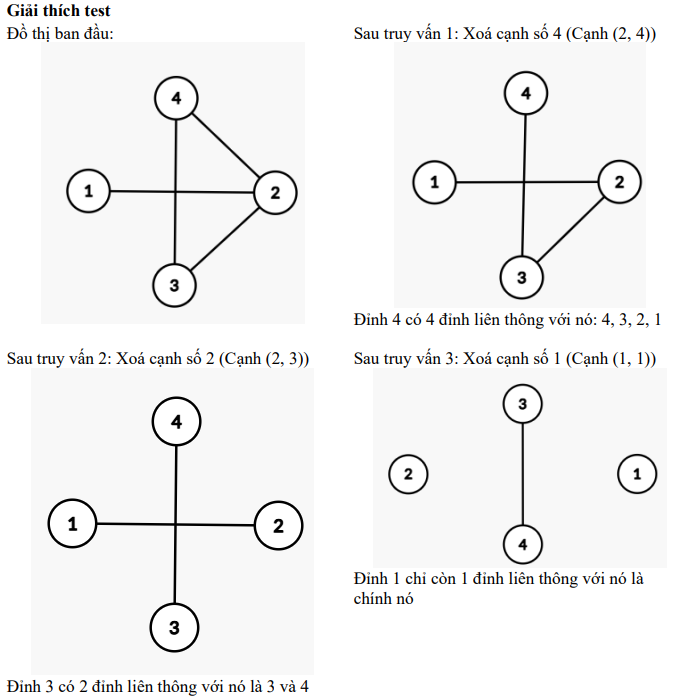
* Dòng đầu tiên ghi 3 số N, M, Q. Với N là số đỉnh, M là số cạnh, Q là số truy vấn
* Tiếp theo là M dòng ghi M cạnh của đồ thị
* Tiếp theo là Q dòng, mỗi dòng gồm 2 số x và y. Trong đó, x là số thứ tự cạnh bị xoá, y là đỉnh cần kiểm tra xem TPLT chứa nó còn lại bao nhiêu đỉnh.

**Ouput.** Với mỗi truy vấn, in ra số đỉnh thuộc thành phần liên thông của đỉnh y trên 1 dòng

**Giới hạn.** 1 ≤ N ≤ 105, 1 ≤ M ≤ 5.105, 1 ≤ Q ≤ 105

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4 4 3  1 2  2 3  3 4  2 4  4 4  2 3  1 1 | 4  2  1 |

****

# **BÀI 27. DSAKT077. TRÒ CHƠI GHÉP CHỮ**

Cho trước một danh sách M từ, gọi là từ điển, mỗi từ là một chuỗi không quá 25 chữ cái thường, không có khoảng trống. Sau đó lần lượt cho các lưới kích thước N \* N các chữ cái viết thường, với 2 ≤ N ≤ 8.

Hãy xác định xem từ một lưới như vậy có thể ghép thành các từ nào trong từ điển M từ ban đầu. Quy tắc ghép chữ từ lưới là chỉ được ghép theo chiều ngang, chiều dọc hoặc đường chéo và mỗi vị trí trên lưới chỉ được dùng một lần.

Ngoài ra, có một ngoại lệ bổ sung là với ô trên lưới có chữ cái **q** thì được xử lý như một cặp chữ cái **qu** khi ghép từ.

**Input**

* Dòng đầu tiên chứa số M là số từ của từ điển (1 ≤ M ≤ 200).
* Tiếp theo là M dòng ghi M từ.
* Tiếp theo là vô hạn các lưới. Mỗi lưới gồm:
  + Dòng đầu tiên của lưới ghi số N là kích thước lưới (2 ≤ N ≤ 8).
  + N dòng tiếp theo, mỗi dòng gồm N chữ cái (từ a đến z).
* Lưới cuối cùng có N = 0.

**Output**

* Với mỗi lưới N\*N đã cho, ghi ra các từ trong từ điển có thể viết ra được từ lưới đó, mỗi từ trên một dòng và sắp xếp theo thứ tự từ điển.
* Sau đó viết tiếp một dòng có dấu – để thông báo kết thúc output của lưới đang xét.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  april  purple  quilt  5  rprit  ahqln  ietep  zrysg  ogwey  3  pel  aup  bcr  0 | april  quilt  -  purple  - |

**Lưu ý:**

* 1. Đây là bài toán khó, các bạn GIỎI có thể thử sức
  2. Bài toán này phải áp dụng khá nhiều kỹ thuật nâng cao. Ngoài DFS thuần tuý, các bạn còn phải thành thạo dp on tree, trie thì mới AC được
  3. Bài này được xây dựng theo ICPC style, chỉ có 1 test duy nhất và là test rất chặt, buộc lời giải phải chuẩn. Không có subtask cho nên cảm thấy khó quá, các bạn có thể bỏ qua làm bài khác.

# **BÀI 28. DSAKT121. DUYỆT ĐỒ THỊ**

Cho đồ thị vô hướng G=<V, E> được biểu diễn dưới dạng danh sách cạnh. Hãy viết thuật toán **duyệt theo chiều sâu** bắt đầu tại đỉnh u.

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm |E| +1 dòng:
  + Dòng đầu tiên đưa vào ba số |V|, |E|, u lần lượt là số đỉnh và số cạnh của đồ thị, u là đỉnh xuất phát;
  + |E| dòng tiếp theo đưa vào các bộ đôi (u, v) tương ứng với một cạnh của đồ thị.
* T, |V|, |E| thỏa mãn ràng buộc: 1 ≤ T ≤ 200; 1 ≤ |V| ≤ 103; 1 ≤ |E| ≤ |

**Output:**Đưa ra danh sách các đỉnh được duyệt theo thuật toán DFS(u) của mỗi test theo khuôn dạng của ví dụ dưới đây.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  6 9 5  1 2  1 3  2 3  2 4  3 4  3 5  4 5  4 6  5 6 | DFS(5) = 5 3 1 2 4 6 |

# **BÀI 29. DSAKT122. ĐỒ THỊ LIÊN THÔNG – 1**

Cho đồ thị vô hướng liên thông G = <V, E>  được biểu diễn dưới dạng danh sách cạnh. Hãy liệt kê các đỉnh u theo thứ tự từ nhỏ đến lớn sao cho nếu bỏ u và các cạnh nối với u ra khỏi đổ thị thì đồ thị không còn liên thông nữa.

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng:
  + Dòng đầu tiên đưa vào hai số |V|, |E| tương ứng với số đỉnh và số cạnh;
  + Dòng tiếp theo đưa vào các bộ đôi (x, y) tương ứng với một cạnh của đồ thị.
* T, |V|, |E| thỏa mãn ràng buộc: 1 ≤ T ≤ 100; 1 ≤ |V| ≤ 103; 1 ≤ |E| ≤ |

**Output:** Với mỗi test cần đưa ra hai dòng

* Dòng đầu tiên ghi số lượng đỉnh thỏa mãn điều kiện
* Dòng thứ hai liệt kê các đỉnh theo thứ tự từ nhỏ đến lớn, mỗi đỉnh cách nhau một khoảng trống.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  5 5  1 2 1 3 2 3 2 5 3 4 | 2  2 3 |

# **BÀI 30. DSAKT123. ĐỒ THỊ LIÊN THÔNG – 2**

Cho đồ thị vô hướng liên thông G = <V, E>  được biểu diễn dưới dạng danh sách cạnh. Hãy liệt kê tất cả các cạnh (u,v) theo thứ tự từ điển thỏa mãn điều kiện: Nếu bỏ cạnh (u,v) ra khỏi đồ thị thì đồ thị không còn liên thông nữa.

**Input:**

* Dòng đầu tiên đưa vào T là số lượng bộ test.
* Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng:
  + Dòng đầu tiên đưa vào hai số |V|, |E| tương ứng với số đỉnh và số cạnh;
  + Dòng tiếp theo đưa vào các bộ đôi (x, y) tương ứng với một cạnh của đồ thị.
* T, |V|, |E| thỏa mãn ràng buộc: 1 ≤ T ≤ 100; 1 ≤ |V| ≤ 103; 1 ≤ |E| ≤ |

**Output:** Với mỗi test cần đưa ra hai dòng

* Dòng đầu tiên là số lượng cạnh thỏa mãn điều kiện
* Dòng thứ hai lần lượt đưa ra danh sách các cạch thỏa mãn theo thứ tự từ điển. Mỗi cạnh cách nhau một khoảng trống.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  5 5  1 2 1 3 2 3 2 5 3 4 | 2  (2 5) (3 4) |

# **BÀI 31. HÌNH CHỮ NHẬT NHỎ NHẤT**

Cho N điểm trên mặt phẳng Oxy và M đoạn thẳng nối các điểm này với nhau. Nhiệm vụ của bạn là cần tìm ra 1 hình chữ nhật có các cạnh song song với 2 trục toạ dộ Ox và Oy, có chu vi nhỏ nhất và che phủ trọn vẹn được 1 tập hợp các điểm liên thông với nhau và tính chu vi của hình chữ nhật đó.

**Lưu ý:** Trường hợp tìm được hình chữ nhật bao phủ là 1 đoạn thẳng (Với các điểm tạo thành đường thẳng song song với 1 trong 2 trục, chiều còn lại bằng 0) thì kết quả ấy vẫn được công nhận. Đáp án là 2 lần độ dài đoạn thẳng đó.

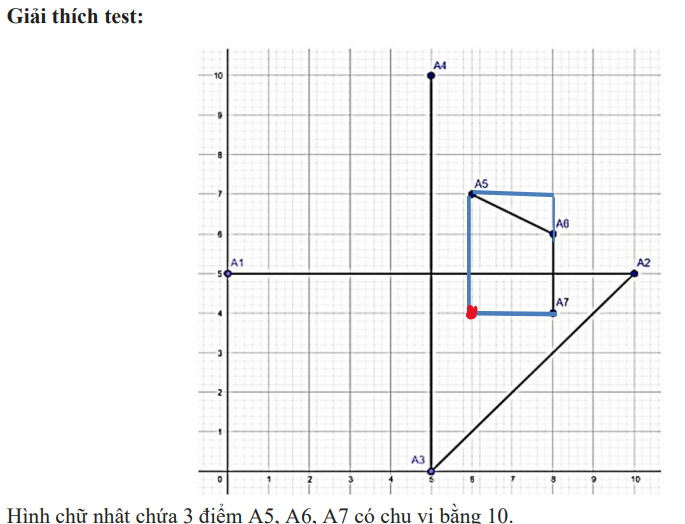
**Input:**

* Dòng đầu tiên gồm hai số nguyên N và M (2 ≤ N, M ≤ 100 000).
* N dòng tiếp theo, mỗi dòng gồm 2 số nguyên X[i] và Y[i] mô tả tọa độ của một điểm. Các tọa độ nằm trong phạm vi [0, 109].
* Theo sau đó là M dòng, mỗi dòng gồm 2 số nguyên u và v, mô tả một cạnh nối điểm u tới điểm v.

**Output:** In ra chu vi của hình chữ nhật nhỏ nhất tìm được

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 7 5  0 5  10 5  5 0  5 10  6 7  8 6  8 4  1 2  2 3  3 4  5 6  7 6 | 10 |



# **BÀI 32. CÂY KHUNG NHỎ NHẤT**

Trong đồ thị vô hướng, liên thông có N đỉnh, M cạnh với trọng số không âm, cây khung nhỏ nhất (minimum spanning tree) là đồ thị con chứa vừa đủ N-1 cạnh, vẫn đảm bảo tính liên thông và tổng trọng số nhỏ nhất có thể.

Bài toán đặt ra là hãy tìm và loại bỏ duy nhất 1 cạnh sao cho giá trị tổng trọng số của cây khung nhỏ nhất là lớn nhất có thể. Giả sử đồ thị luôn đảm bảo nếu loại bỏ 1 cạnh bất kỳ thì vẫn liên thông.

**Input.**

* Dòng đầu ghi 2 số N và M (3 ≤ N ≤ 105 ; 3 ≤ M ≤ 106). Các đỉnh đánh số từ 1 đến N.
* M dòng tiếp theo, mỗi dòng ghi 3 số u, v, w, tương ứng là đỉnh đầu, đỉnh cuối, và trọng số từng cạnh. (1 ≤ u < v ≤ N ; 1 ≤ w ≤ 106).

**Output.** Ghi ra tổng trọng số của cây khung nhỏ nhất sau khi đã xóa 1 cạnh như mô tả trên.

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 7  2 5 8  1 3 19  4 5 9  1 5 15  1 2 14  3 4 16  2 4 15 | 54 |
| 4 5  2 3 5  1 2 2  1 3 4  1 4 2  3 4 3 | 10 |

# **BÀI 33. KIỂM TRA ĐỒ THỊ Ơ-LE**

Cho đồ thị vô hướng liên thông G = <V, E> được biểu diễn dưới dạng danh sách cạnh. Nhiệm vụ của bạn là đưa ra “Yes” nếu đồ thị là nửa Euler, ngược lại đưa ra “No”.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất là bộ đôi V, E; phần thứ 2 đưa vào |E| cạnh của đồ thị G.
* Số đỉnh của đồ thị không quá 1000, số cạnh của đồ thị không quá 10000.

**Output:** Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  5 7  1 2 1 3 2 3 2 5 3 4 3 5 4 5 | Yes |

**(Thi chính thức D21CN, ca 8h, tái hiện)**

# **BÀI 33. ĐƯỜNG KÍNH CỦA CÂY**

Cho đồ thị vô hướng liên thông có N đỉnh và N – 1 cạnh. Tính đường đi dài nhất giữa 2 đỉnh bất kỳ trong đồ thị. **(Thi chính thức D21CN, ca 13h, tái hiện)**

**Input.**

* Dòng đầu tiên là số N (1 ≤ N ≤ 105) là số đỉnh của đồ thị
* N dòng tiếp theo, mỗi dòng là bộ đôi 2 số x, y mô tả cạnh của đồ thị

**Ouput.**

In ra độ dài đường đi dài nhất giữa 2 đỉnh bất kỳ trên đồ thị

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5  1 2  2 3  2 4  1 5 | 3 |

**Giải thích test.** Có 2 đường đi dài nhất: 5 🡪 1 🡪 2 🡪 3, 5 🡪 1 🡪 2 🡪 4. Chúng đều có độ dài là 3.

# **BÀI 34. DSATest28. HỆ THỐNG GIAO THÔNG**

Mạng lưới giao thông ở 1 nước bao gồm N thành phố (đánh số từ 1 đến N) và N - 1 đường nối các thành phố với nhau. Có một đường đi duy nhất giữa mỗi cặp thành phố và mỗi con đường có một độ dài xác định.

Nhiệm vụ của bạn là với mỗi K cặp thành phố cho trước, tìm độ dài của con đường ngắn nhất và dài nhất giữa 2 thành phố này.

**Input:**

* Dòng đầu tiên chứa số nguyên N (2 ≤ N ≤ 105).
* N - 1 dòng tiếp theo chứa 3 số nguyên A, B, C cho biết có một con đường độ dài C giữa thành phố A và thành phố B (1 ≤ C ≤ 106).
* Dòng tiếp theo chứa số nguyên K là số lượng truy vấn (1 ≤ K ≤ 105).
* K dòng tiếp theo, mỗi dòng gồm 2 số nguyên U và V.

**Output:**

Với mỗi truy vấn, in ra hai số nguyên là độ dài đường đi ngắn nhất và dài nhất tìm được.

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5  2 3 100  4 3 200  1 5 150  1 3 50  3  2 4  3 5  1 2 | 100 200  50 150  50 100 |
| 7  3 6 4  1 7 1  1 3 2  1 2 6  2 5 4  2 4 4  5  6 4  7 6  1 2  1 3  3 5 | 2 6  1 4  6 6  2 2  2 6 |