**A. Đồ thị vô hướng**

1. Chu trình Euler và chu trình Hamilton.

Chu trình Euler là chu trình đi qua tất cả các cạnh, mỗi cạnh đúng một lần.

Chu trình Hamilton là chu trình đi qua tất cả các đỉnh, mỗi đỉnh đúng một lần. Xét các đồ thị được cho bởi bốn tập cạnh sau:

0-1 0-2 0-3 1-3 1-4 2-5 2-9 3-6 4-7 4-8 5-8 5-9 6-7 6-9 7-8

0-1 0-2 0-3 1-3 0-3 2-5 5-6 3-6 4-7 4-8 5-8 5-9 6-7 6-9 8-8

0-1 1-2 1-3 0-3 0-4 2-5 2-9 3-6 4-7 4-8 5-8 5-9 6-7 6-9 7-8

4-1 7-9 6-2 7-3 5-0 0-2 0-8 1-6 3-9 6-3 2-8 1-5 9-8 4-5 4-7

Đồ thị nào có chu trình Euler? Đồ thị nào có chu trình Hamilton?

Giải

Đồ thị vô hướng là đồ thị Euler nếu và chỉ nếu:

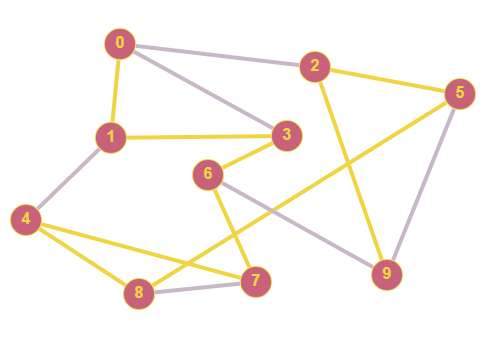
* Bậc của mọi đỉnh là chẵn
* Tất cả các đỉnh có bậc lớn hơn 0 thuộc cùng một thành phần liên thông

0-1 0-2 0-3 1-3 1-4 2-5 2-9 3-6 4-7 4-8 5-8 5-9 6-7 6-9 7-8

Có:

* Đồ thị này không có chu trình Euler

Đồ thị có chu trình Hamilton: 0⇒1⇒3⇒6⇒7⇒4⇒8⇒5⇒9⇒2⇒0

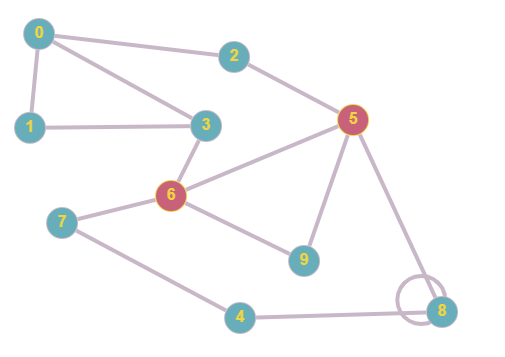


0-1 0-2 0-3 1-3 0-3 2-5 5-6 3-6 4-7 4-8 5-8 5-9 6-7 6-9 8-8

Có:

* Đồ thị này không có chu trình Euler

Đồ thị này không có chu trình Hamilton vì xóa 2 đỉnh (5 và 6) xuất hiện 3 thành phần liên thông:

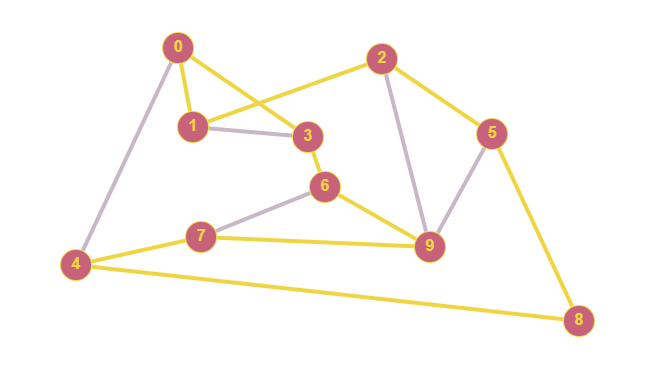


0-1 1-2 1-3 0-3 0-4 2-5 2-9 3-6 4-7 4-8 5-8 5-9 6-7 6-9 7-8

Có:

* Đồ thị này không có chu trình Euler

Đồ thị có chu trình Hamilton: 0⇒1⇒2⇒5⇒8⇒4⇒7⇒9⇒6⇒3⇒0

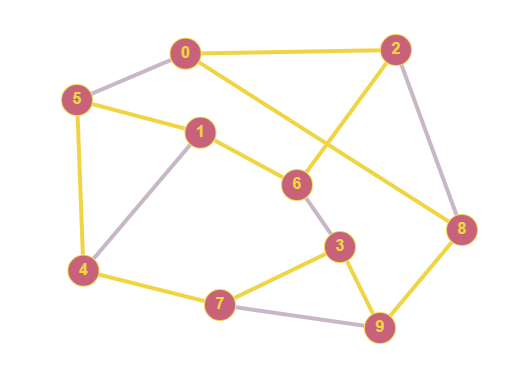


4-1 7-9 6-2 7-3 5-0 0-2 0-8 1-6 3-9 6-3 2-8 1-5 9-8 4-5 4-7

Có:

* Đồ thị này không có chu trình Euler

Đồ thị có chu trình Hamilton: 0⇒2⇒6⇒1⇒5⇒4⇒7⇒3⇒9⇒8⇒0



2. Đếm đồ thị

Có bao nhiêu đồ thị vô hướng khác nhau có V đỉnh và E cạnh (không có cạnh song song)?

Giải

Số cạnh lớn nhất của đồ thị là

Đồ thị vô hướng có V đỉnh thì sẽ có tối đa cạnh

Chọn E cạnh trong cạnh có thể có của đồ thị: cách chọn

* Có thể tạo ra đồ thị vô hướng từ V đỉnh và E cạnh

3. Phát hiện cạnh song song Thiết kế một thuật toán tuyến tính để đếm số cạnh song song trong một đồ thị

Giả sử input là danh sách cạnh, m là số cạnh

vector<int> edges[m]

Procedure addEdges() {

for (int i = 0; i < m; i++) {

edges[x].push\_back(y);

trong đó x là đỉnh hiện tại, y là đỉnh kết thúc

}

}

Procedure int solution() {

int cnt = 0;

for (các đỉnh trong đồ thị) {

Nếu mỗi đỉnh hiện tại tồn tại đỉnh x mà trong danh sách của edge[x] cũng tồn

tại định hiện tại thì cnt++;

}

return cnt;

}

4. Chu trình lẻ

Chứng minh rằng một đồ thị là đồ thị hai mầu (bipartite) khi và chỉ khi nó không chứa chu trình độ dài lẻ. Đồ thị hai mầu là đồ thị mà có thể dùng hai mầu để tô mỗi đỉnh một mầu sao cho không có cạnh nào nối giữa hai đỉnh cùng mầu.

Gợi ý: chứng minh bằng phản chứng.

Giải

Chứng minh bằng phản chứng:

1. Giả sử G là đồ thị bipartite => G không chứa chu trình lẻ

Nếu G có chu trình lẻ

Do C chỉ được tô bằng 2 màu

* Các đỉnh lẻ sẽ được tô bằng 1 màu nhưng và là 2 đỉnh kề nhau có cùng màu
* Vô lý

1. Giả sử G không chứa chu trình lẻ => G là đồ thị bipartite

Chọn 1 đỉnh r làm gốc và tô nó màu đỏ. ∀ x ∈ V sẽ được tô màu đỏ nếu đường đi ngắn nhất từ x tới r có số ca.nh chẵn. Trái lại tô x màu xanh.

Trái lại giả sử x và y là 2 đỉnh của cạnh (x,y) nào đó được tô cùng màu (cùng chẵn hoặc lẻ)

* Vô lý => dpcm

5. Biconnected

Một đồ thị được gọi là biconnected nếu mỗi cặp đỉnh đều được nối với nhau bởi hai đường đi không giao nhau. Trong đồ thị liên thông, điểm articulation là đỉnh mà khi xóa nó và các cạnh kề sẽ làm đồ thị mất tính liên thông. Hãy chứng minh rằng một đồ thị bất kì mà không có điểm articulation là đồ thị biconnected.

Gợi ý: cho một cặp đỉnh s và t và một đường đi nối giữa chúng. Hãy sử dụng dữ kiện rằng không có đỉnh nào trên đường đi đó là điểm articulation để xây dựng hai đường đi không giao nhau nối s và t.

Giải

Giả sử s là điểm articulation của G

* Trong G – s, đỉnh t sẽ nằm trong một thành phần khác với các đỉnh còn lại.
* Trong G – s có một đỉnh x trong một thành phần khác với v
* s nằm trên mọi đường đi t – x

P là đường đi ngắn nhất t – x trong G. Vì P chứa s

* Vô lý
* Dpcm