



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

THUẬT TOÁN ỨNG DỤNG

Thực hành
Buổi 5: Đồ thị

NỘI DUNG

- 01. Tìm điểm khớp và cạnh cầu(Articulation Points and Bridges)
- 02. Make span schedule
- 03. Hệ thống xe buýt đô thị(Inter City Bus System)
- 04. Tìm thành phần liên thông mạnh (Strongly Connected Component)

01. Tìm điểm khớp và cạnh cầu (Articulation Points and Bridges)

- 1.1 Lịch sử bài toán
- 1.2 Phát biểu bài toán
- 1.3 Ứng dụng
- 1.4 Thuật giải và cài đặt

1.1 Lịch sử bài toán

- Trên đồ thị vô hướng/có hướng, liên thông/liên thông mạnh:
 - Khớp là đỉnh mà khi loại nó khỏi đồ thị thì số thành phần liên thông/liên thông mạnh của đồ thị tăng lên.
 - Cầu là cạnh mà khi loại nó khỏi đồ thị thì số thành phần liên thông/ liên thông mạnh của đồ thị tăng lên.

1.2 Phát biểu bài toán

- Cho đồ thị vô hướng n đỉnh và m cạnh.
 - Tìm số lượng khớp (articulation point)
 - Tìm số lượng cầu (bridge) của đồ thị.

Bridges

Input

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

Output

4 3

1.3 Ứng dụng

- Trong lĩnh vực mạng thì việc kiểm tra và tìm ra các điểm khớp và cầu có ý nghĩa quan trọng, cắt được những điểm này thì hệ thống mạng sẽ an toàn hơn.

1.4 Thuật giải và cài đặt

- Thuật toán 1: Duyệt tất cả
- Thuật toán Tarjan

1.4 Thuật giải và cài đặt

Thuật toán 1: Duyệt tất cả

- Duyệt qua tất cả các đỉnh của đồ thị, với mỗi đỉnh kiểm tra xem có phải là khớp của đồ thị hay không
- Sử dụng thuật BFS hoặc DFS để kiểm tra
- Độ phức tạp $O(n * m)$

1.4 Thuật giải và cài đặt

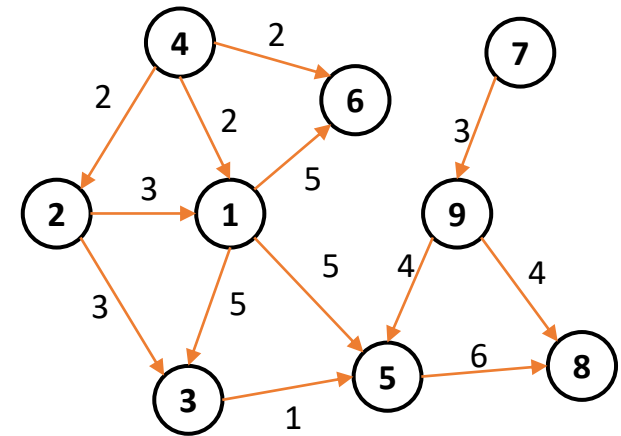
Thuật toán Tarjan

- Duyệt đồ thị theo thứ tự DFS, ta liệt kê được thứ tự các đỉnh được duyệt, ta gọi là cây DFS.
- Với mỗi đỉnh ta cần tính 2 mảng
 - $num[u]$: Thời gian đếm thăm đỉnh u theo thứ tự duyệt DFS.
 - $low[u]$: Thời gian nhỏ nhất mà đỉnh u và các đỉnh con của u có thể đến thăm được trong cây DFS.
- **Khớp** là các đỉnh u mà có $num[u] \leq low[v]$ trong đó v là con trực tiếp của u trong cây DFS.
- Đối với đỉnh root của cây DFS thì phải có thêm điều kiện có ít nhất 2 con trong cây DFS.
- **Độ phức tạp** : $O(n + m)$.

2. Make span schedule

- Một dự án có n nhiệm vụ $1, \dots, N$. Nhiệm vụ i có thời lượng $d(i)$ phải hoàn thành ($i=1, \dots, n$).
- Có các ràng buộc về quyền ưu tiên giữa các nhiệm vụ được biểu diễn bởi một tập hợp các cặp Q : đối với mỗi cặp (i, j) trong Q , nhiệm vụ j không thể được bắt đầu trước khi hoàn thành nhiệm vụ i .
- Tính thời gian hoàn thành sớm nhất của dự án.
- Đầu vào
 - Dòng 1: chứa n và m ($1 \leq n \leq 10^4, 1 \leq m \leq 200000$)
 - Dòng 2: chứa $d(1), \dots, d(n)$ ($1 \leq d(i) \leq 1000$)
 - Dòng $i+3$ ($i=1, \dots, m$): chứa i và j : không thể bắt đầu thực thi tác vụ j trước khi hoàn thành tác vụ i
- Đầu ra
 - Ghi thời gian hoàn thành sớm nhất của dự án.

Make Span Schedule



- Thuật giải
 - L là danh sách TOPO các nút của G
 - $F[u]$: thời điểm sớm nhất mà nhiệm vụ u có thể bắt đầu
 - Khám phá L từ trái sang phải, với mỗi nút u:
 - $\text{makepan} = \max(\text{makespan}, F[u] + d[u])$
 - Với mỗi cung (u,v) , cập nhật $F[v] = \max(F[v], F[u] + d[u])$

03. Hệ thống xe buýt đô thị (Inter City Bus System)

- 3.1 Lịch sử bài toán
- 3.2 Phát biểu bài toán
- 3.3 Ứng dụng
- 3.4 Thuật giải và cài đặt

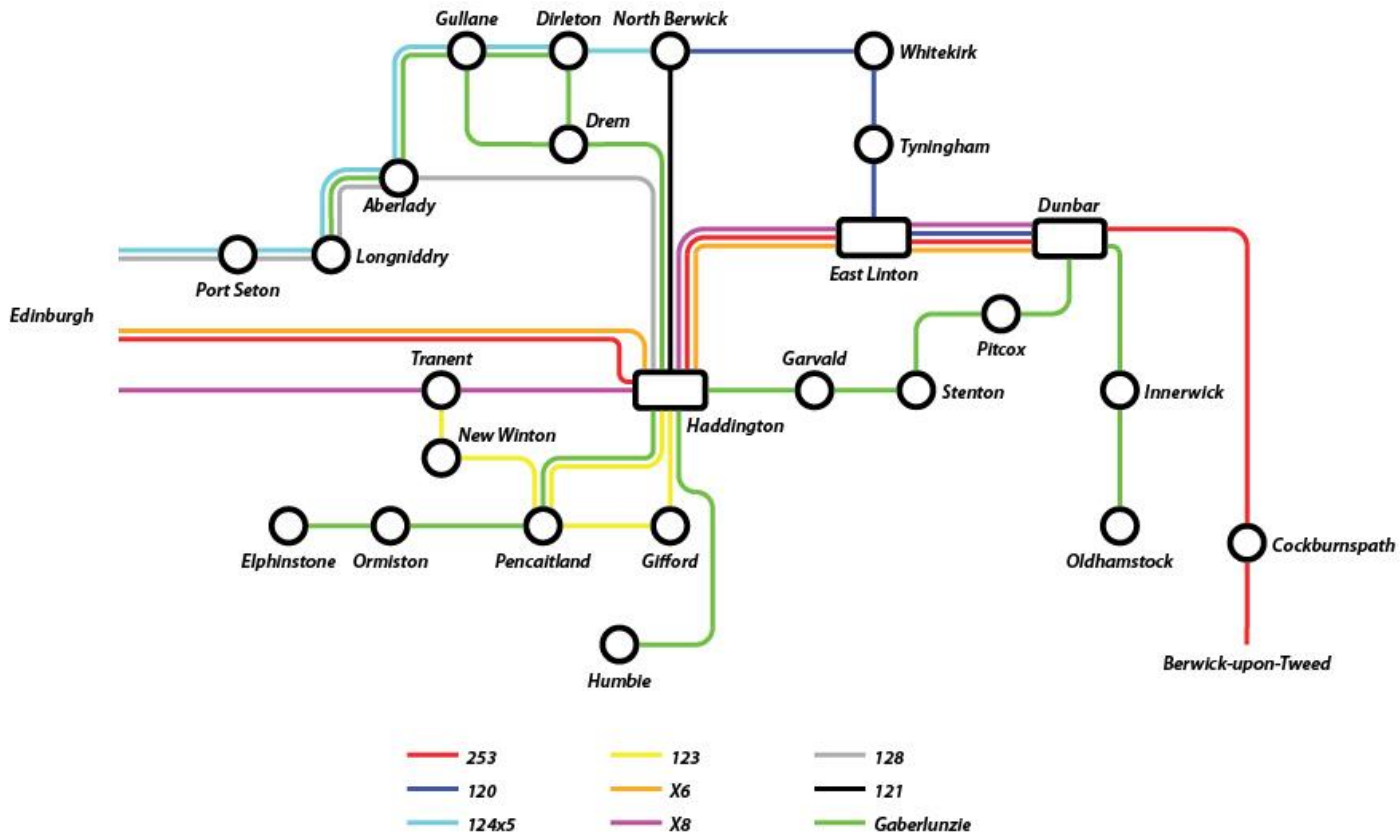
3.1 Lịch sử bài toán

- Khi các đô thị phát triển, dân cư ngày càng đông đúc thì phương tiện giao thông công cộng trở thành phương tiện chính trong thành phố.
- Vì vậy bài toán bố trí tuyến xe buýt hiệu quả và giảm thiểu chi phí được đặt ra.
- Khi các tuyến xe được bố trí thì việc lựa chọn tuyến đi để tối ưu chi phí cũng là một bài toán cần giải quyết

3.2 Phát biểu bài toán

- Cho n thị trấn được đánh số từ 1 tới n .
- Có k con đường hai chiều nối giữa các thị trấn.
- Ở thị trấn thứ i sẽ có một tuyến bus với giá vé là c_i và đi được quãng đường tối đa là d_i .
- Tìm chi phí tối thiểu để đi từ thị trấn 1 tới thị trấn n .

3.2 Phát biểu bài toán



Inter-City Bus

Input

6 6

10 2

30 1

50 1

20 3

30 1

20 1

1 2

1 3

1 5

2 4

2 5

4 6

Output

30

Giải thích:

-Lên buýt 1 từ thành phố 1 đến thành phố 4 mất 10 đồng

-Lên buýt 4 từ thành phố 4 đến thành phố 6 mất 20 đồng

Tổng cộng mất $10 + 20 = 30$ đồng

3.3 Ứng dụng

- Tìm đường đi tối ưu trong bài toán tối ưu chi phí cho bài toán vận chuyển



3.4 Thuật giải và cài đặt

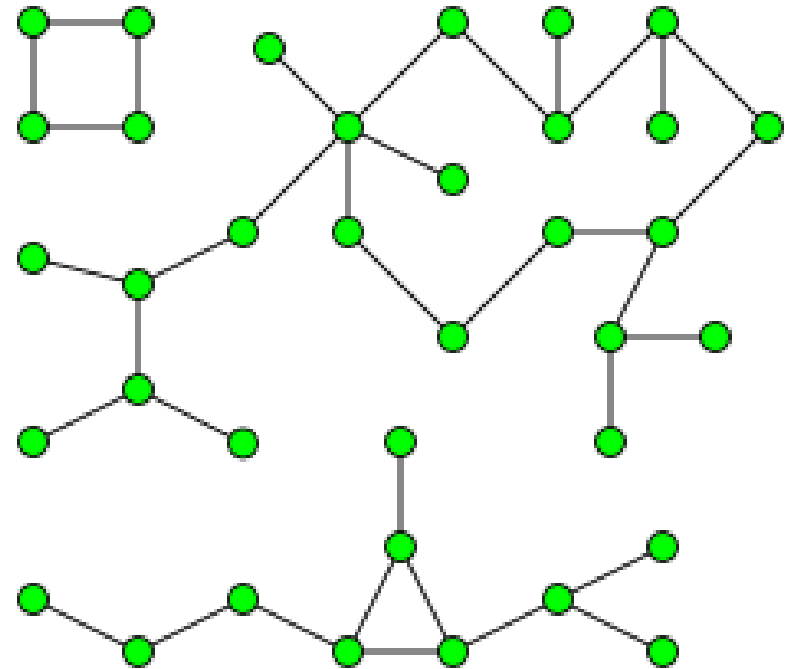
- **Bước 1** : Tính khoảng cách di chuyển ngắn nhất của tất cả các cặp đỉnh $u; v$ bằng thuật toán BFS. Lưu vào mảng $dist[u][v]$
- **Bước 2** : Tạo một đồ thị mới một chiều trong đó đỉnh u được nối tới đỉnh v khi $dist[u][v] \leq d[u]$ và cạnh này có trọng số là $c[u]$
- **Bước 3** : Tìm đường đi ngắn nhất từ 1 tới n trên đồ thị mới được tạo ra bằng thuật toán Dijkstra.
- **Độ phức tạp** thuật toán $O(n^2)$

04. Tìm thành phần liên thông (Connected Components)

- 4.1 Lịch sử bài toán
- 4.2 Phát biểu bài toán
- 4.3 Ứng dụng
- 4.4 Thuật giải và cài đặt

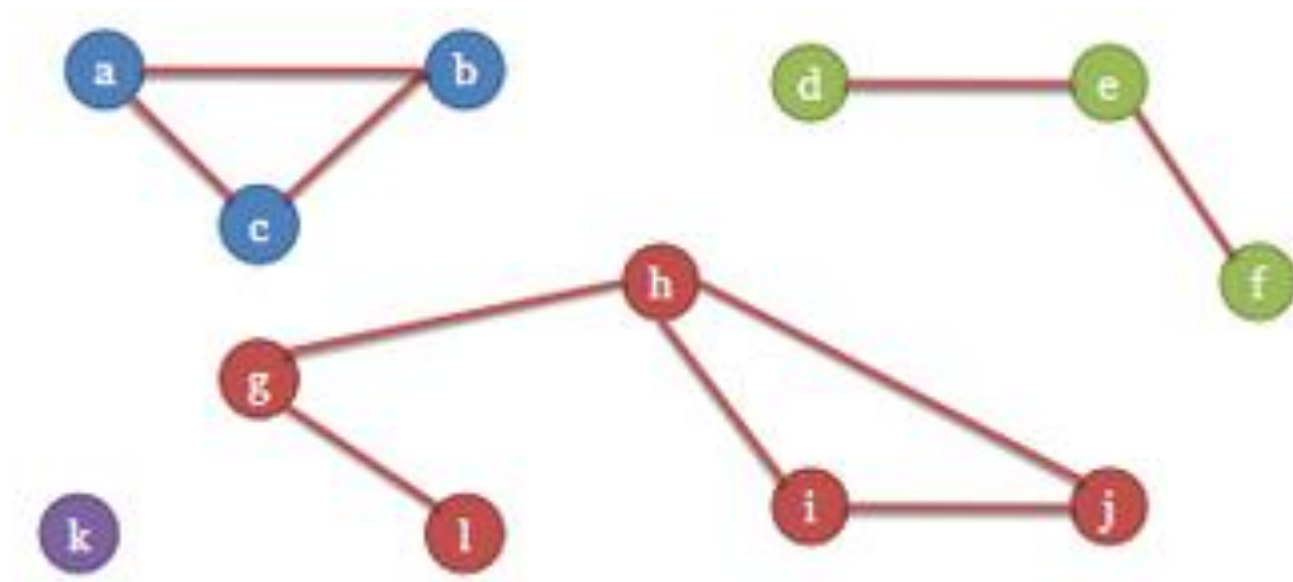
4.1 Lịch sử bài toán

- Thành phần liên thông chỉ ra sự kết nối giữa các thành phần của đồ thị.
- Bài toán được đề cập đến trong các vấn đề thực tiễn khi muốn tìm sự kết nối giữa các thành phần trong đồ thị.



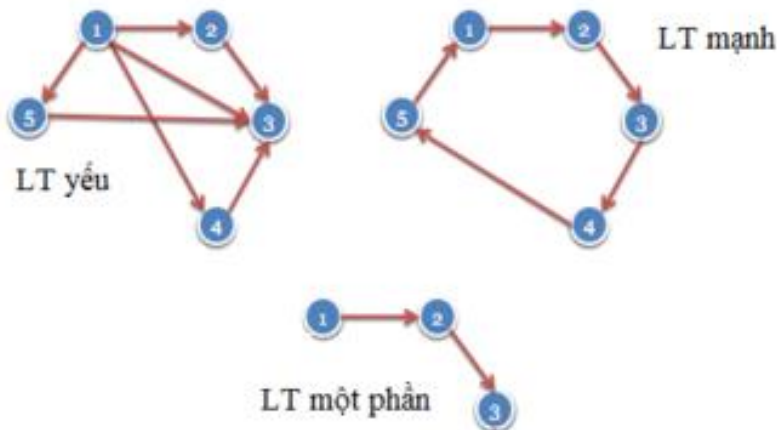
4.2 Phát biểu bài toán

- Cho một đồ thị có n đỉnh và m cạnh 2 chiều
- Tính số lượng thành phần liên thông của đồ thị



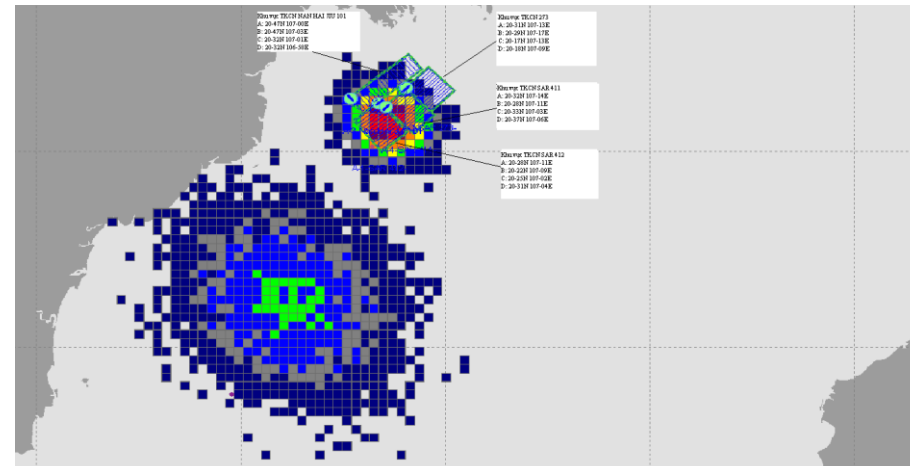
4.2 Phát biểu bài toán

- Cho một đồ thị có hướng $G=(V,E)$ trong đó $V=\{1, \dots, N\}$ là số nút và tập hợp E có M cung. Tính số thành phần liên thông mạnh của G
- Đầu vào
 - Dòng 1: 2 số nguyên dương N và M ($1 \leq N \leq 10^5$, $1 \leq M \leq 10^6$)
 - Dòng $i+1$ ($i=1, \dots, M$): chứa 2 số nguyên dương u và v là điểm cuối của cung thứ i
- Đầu ra
 - Viết số thành phần liên thông mạnh của G



4.3 Ứng dụng

- BÀI TOÁN "BẢN ĐỒ CÁC VÙNG ĐẢO"
- Khi tiến hành khảo sát các đảo ở một vùng biển, người ta ghi kết quả khảo sát lại thành một bản đồ nhị phân
 - Trong đó số 0 cho biết biển và số 1 là đất liền
 - Thể hiện trên 1 bản cờ ở dạng kẻ lưới
- Bài toán bản đồ các vùng đảo,
- là bài toán ứng dụng thực tế của việc tìm thành phần liên thông
- là tiền đề cho việc thực hiện một số trò chơi nổi tiếng như: Minesweeper
- Trong bài toán tìm kiếm cứu nạn trên các vùng biển.
- Mạng xã hội: giới thiệu bạn bè gần gũi



Strongly Connected Component

Input

8 13

1 2

1 8

2 3

2 6

3 6

4 3

4 6

5 4

6 5

7 1

7 2

7 6

8 7

Output

3

4.4 Thuật giải và cài đặt

- Chạy DFS trên $G \rightarrow$ tính thời gian kết thúc $f(v)$ của mỗi nút v của G
- Dựng đồ thị dư G^T của G
- Chạy DFS trên G^T : các nút được xét theo thứ tự f giảm dần
 - Mỗi lần chạy $\text{DFS}(u)$ sẽ truy cập tất cả các nút của thành phần được kết nối mạnh có chứa u



25 YEARS ANNIVERSARY
SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

**Thank you
for your
attentions!**



soict.hust.edu.vn/



fb.com/groups/soict

