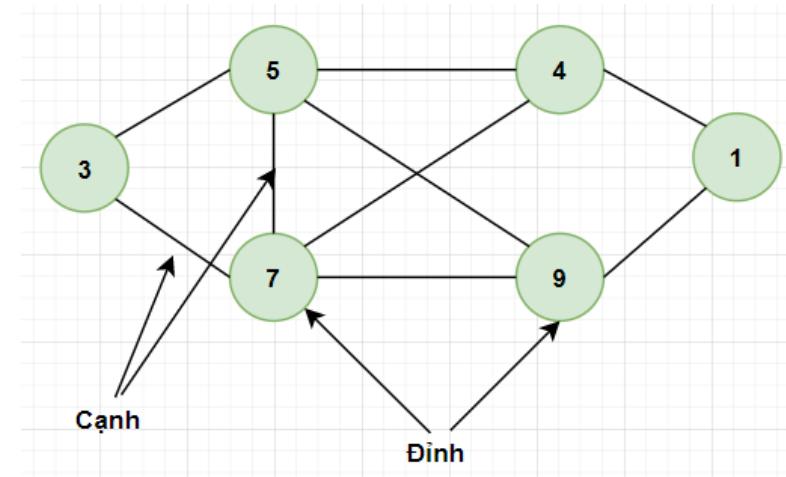


Bài 10.1: Tổng quan về đồ thị

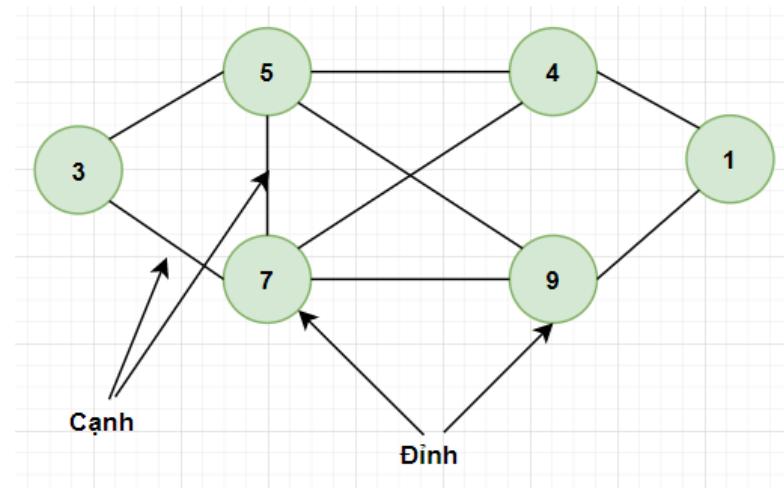
- ✓ Khái niệm
- ✓ Các thao tác trên đồ thị
- ✓ Các cấu trúc dữ liệu biểu diễn đồ thị
- ✓ Ứng dụng của đồ thị

Khái niệm

- Trong khoa học máy tính, một đồ thị là một cấu trúc dữ liệu trừu tượng dùng để triển khai các đồ thị có hướng và vô hướng. Đây là các khái niệm trong lĩnh vực lý thuyết đồ thị của toán học.
- Một cấu trúc dữ liệu đồ thị(graph) bao gồm một tập hữu hạn các đỉnh và các cạnh.
- Một đỉnh hay còn gọi là node, hay điểm là nơi giao nhau của các cạnh. Thường được đặt tên cho dễ phân biệt.
- Một cạnh là đường nối giữa hai đỉnh.



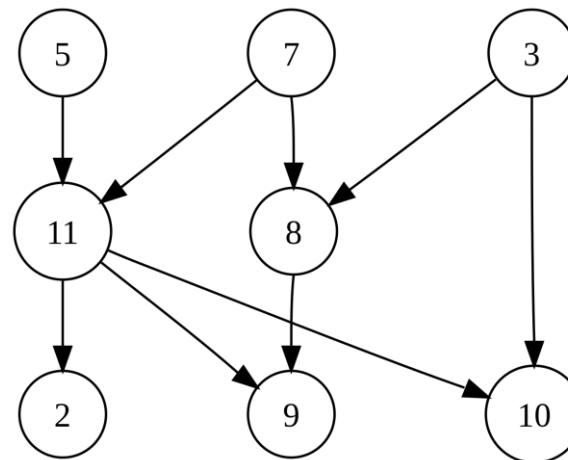
Khái niệm



- Tập các đỉnh của đồ thị thường kí hiệu V (vertice).
- Tập các cạnh của đồ thị thường kí hiệu là E (edge).
- Trong đồ thị trên tập đỉnh là $V = \{3, 7, 5, 4, 9, 1\}$
- Tập cạnh là $E = \{(3, 5), (3, 7), (5, 7), (5, 9), (5, 4), (7, 4), (7, 9), (4, 1), (9, 1)\}$
- Đồ thị $G = \{V, E\}$

Khái niệm

- Trong đồ thị có hướng, cạnh là cặp đỉnh có thứ tự khi biểu diễn bằng hình ảnh thường có mũi tên từ đỉnh bắt đầu đến đỉnh kết thúc.
- Trong đồ thị vô hướng cặp đỉnh không cần có thứ tự.
- Mỗi cạnh thường được gán một vài giá trị biểu diễn các đại lượng như chi phí, độ dài, khả năng...



Các thao tác trên đồ thị

- Thêm đỉnh mới vào đồ thị.
- Xóa đỉnh khỏi đồ thị.
- Thêm cạnh mới vào đồ thị.
- Xóa cạnh khỏi đồ thị.
- Kiểm tra đỉnh kề với một đỉnh.
- Tìm các đỉnh hàng xóm của một đỉnh trong đồ thị.
- Lấy giá trị của đỉnh trong đồ thị.
- Gán giá trị cho đỉnh của đồ thị.
- Lấy giá trị của cạnh của đồ thị.
- Gán giá trị cho cạnh của đồ thị.

Các cấu trúc dữ liệu biểu diễn đồ thị

- Danh sách kề.
- Ma trận kề.
- Ma trận liên kết.

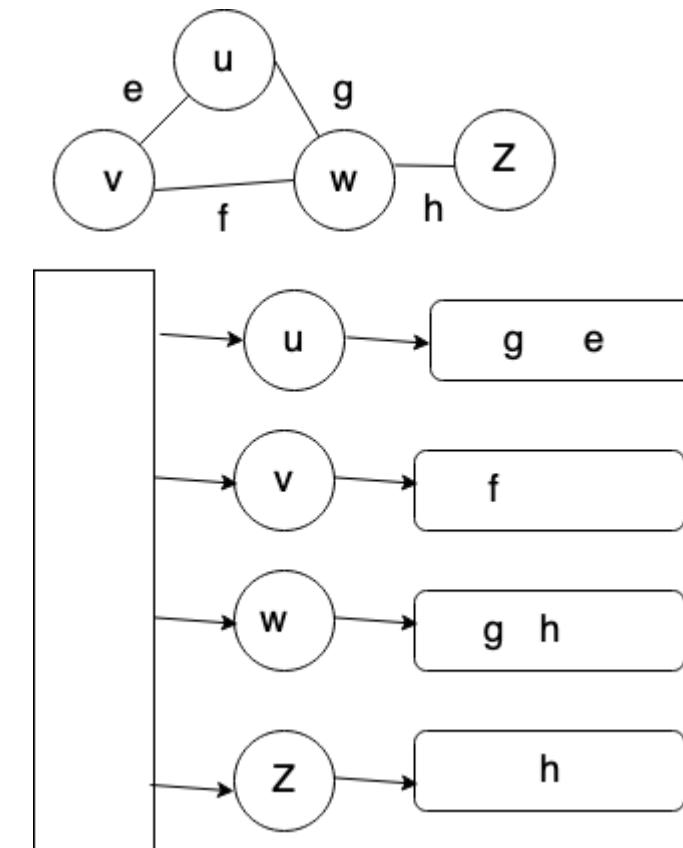
Danh sách kề

- Là một tập hợp của các danh sách không có thứ tự. Dùng để biểu diễn một đồ thị hữu hạn.
- Mỗi đỉnh lưu trữ một danh sách các đỉnh kề hoặc cạnh kề với nó.
- Cấu trúc dữ liệu này cho phép lưu trữ thêm các thông tin bổ trợ tại mỗi đỉnh.

Các triển khai:

- Sử dụng bảng băm liên kết mỗi đỉnh của đồ thị với một mảng các đỉnh kề với nó.
- Sử dụng trực tiếp chỉ số mảng làm đỉnh, mỗi ô trong bảng trả đến một danh sách liên kết chứa các đỉnh kề với nó.
- Cách thứ 3 coi đỉnh, cạnh là các đối tượng. Mỗi đối tượng đỉnh chứa tham chiếu đến danh sách các cạnh kề. Tại mỗi đối tượng cạnh sẽ chứa tham chiếu đến hai đỉnh đầu cuối.

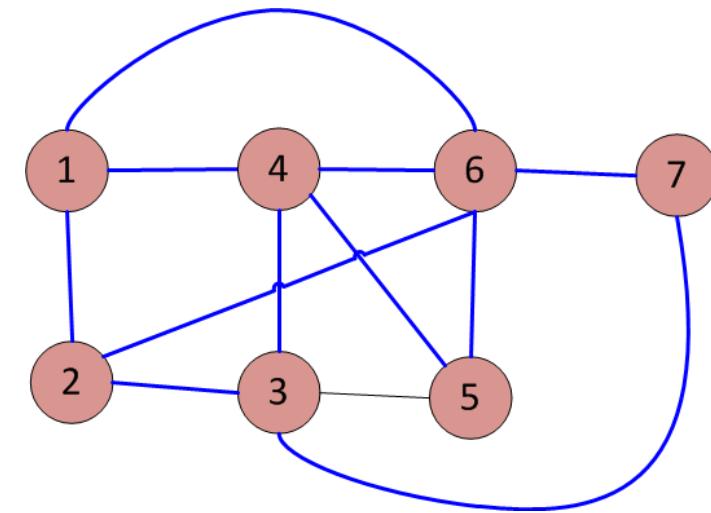
Danh sách kề



Ma trận kề

- Là một ma trận nhị phân được sử dụng để biểu diễn đồ thị vô hướng hoặc có hướng hữu hạn đỉnh.
- Nếu đồ thị G có n đỉnh thì ma trận kề của nó có kích thước $n \times n$. Nếu đồ thị thừa(có ít đỉnh) nên sử dụng danh sách kề để tiết kiệm bộ nhớ cần sử dụng.
- Trong đó các ô $a[i, j]$ trên đường chéo chính là số khuyên tại đỉnh i . Các ô không nằm trên đường chéo chính biểu diễn sự tồn tại của liên kết giữa hai đỉnh i, j .
- Xét đồ thị $G = \{V, E\}$ trong đó các đỉnh $V = \{v_1, \dots, v_n\}$ và các cạnh $E = \{e_1, \dots, e_n\}$. Gọi $A = (a_{ij})$ là ma trận kề của đồ thị G :
 - Nếu tồn tại đường nối giữa đỉnh i và j thì $a_{ij} = 1$.
 - Nếu không tồn tại đường nối giữa hai đỉnh i và j thì $a_{ij} = 0$.
- Với đồ thị vô hướng G , ma trận kề là ma trận nhị phân đối xứng. Tức $a_{ij} = a_{ji}$

Ma trận kề của đồ thị vô hướng



	1	2	3	4	5	6	7
1	0	1	0	1	0	1	0
2	1	0	1	0	0	1	0
3	0	1	0	1	1	0	1
4	1	0	1	0	1	1	0
5	0	0	1	1	0	1	0
6	1	1	0	1	1	0	1
7	0	0	1	0	0	1	0

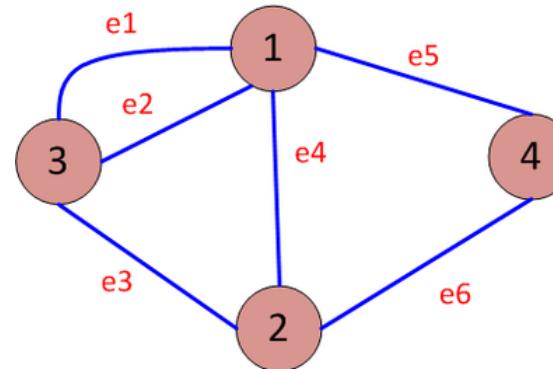
Các đặc điểm

- Ma trận kề không biểu diễn được đồ thị có cạnh song song.
- Ma trận kề của đồ thị có hướng không đối xứng.
- Với đồ thị vô hướng, tổng các phần tử trên hàng i bằng bậc đỉnh i.
- Với đồ thị có hướng, tổng các phần tử trên hàng i bằng bán bậc ra của đỉnh i.
- Ưu điểm: thực hiện kiểm tra đỉnh kề với hai đỉnh u, v chỉ với thời gian $O(1)$.
- Nhược điểm: không phụ thuộc số cạnh của đồ thị, luôn sử dụng n^2 bộ nhớ để lưu trữ ma trận kề.

Ma trận liên kết

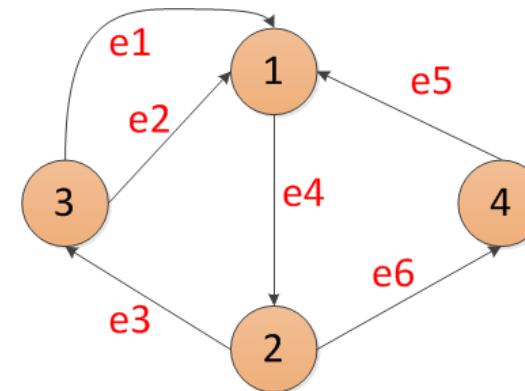
- Là một ma trận nhị phân được sử dụng để biểu diễn đồ thị vô hướng hoặc có hướng hữu hạn đỉnh. Còn được gọi là ma trận liên thuộc.
- Nếu đồ thị G có m đỉnh n cạnh thì ma trận kề của nó có kích thước $m \times n$.
- Trong đó các hàng biểu diễn các đỉnh v_i . Các cột biểu diễn các cạnh e_j .
- Xét đồ thị $G = \{V, E\}$ trong đó các đỉnh $V = \{v_1, \dots, v_n\}$ và các cạnh $E = \{e_1, \dots, e_n\}$. Gọi $A = (a_{ij})$ là ma trận kề của đồ thị G :
 - Nếu đỉnh v_i kề cạnh e_j thì $a_{ij} = 1$.
 - Nếu đỉnh v_i không kề cạnh e_j thì $a_{ij} = 0$.
 - Với đồ thị có hướng, nếu mũi tên từ cạnh cạnh e_j đi ra khỏi đỉnh v_i thì $a_{ij} = 1$.
 - Với đồ thị có hướng, nếu mũi tên từ cạnh cạnh e_j đi vào đỉnh v_i thì $a_{ij} = -1$.

Ma trận liên kết của đồ thi vô hướng



	e1	e2	e3	e4	e5	e6
1	1	1	0	1	1	0
2	0	0	1	1	0	1
3	1	1	1	0	0	0
4	0	0	0	0	1	1

Ma trận liên kết của đồ thi có hướng



	e1	e2	e3	e4	e5	e6
1	-1	-1	0	1	-1	0
2	0	0	1	-1	0	1
3	1	1	-1	0	0	0
4	0	0	0	0	1	-1

Các đặc điểm

- Trong ma trận của đồ thị có hướng, tổng bậc ra của tất cả các đỉnh = tổng bậc vào của tất cả các đỉnh = số cạnh. Bậc là số cạnh nối với đỉnh.
- Trong ma trận của đồ thị vô hướng, tổng bậc của tất cả các đỉnh bằng 2 lần số cạnh.
- Ưu điểm: biểu diễn được đồ thị có cạnh song song; ma trận liên kết tiết kiệm bộ nhớ hơn so với ma trận kề.
- Nhược điểm: biểu diễn phức tạp.

Ứng dụng thực tế của đồ thị

- Google map. Mỗi điểm giao lộ là một đỉnh, đường nối giữa hai giao lộ là cạnh. Từ đó thấy rằng để tìm đường đi ngắn nhất giữa hai điểm A, B chỉ cần tìm đường đi ngắn nhất giữa tất cả các đỉnh đi từ A đến B.
- Mạng xã hội như Facebook. Trong đó mỗi người dùng là một đỉnh. Giữa người dùng và bạn bè họ tồn tại một cạnh để trao đổi thông tin. Gợi ý kết bạn cũng sử dụng lý thuyết đồ thị. Đây là ví dụ về đồ thị vô hướng.
- Mạng internet. Mỗi trang web là một đỉnh, đường link từ trang này tới trang kia là cạnh. Đây là ví dụ về đồ thị có hướng.
- Trong hệ điều hành sử dụng đồ thị để cấp phát tài nguyên. Trong đó mỗi tài nguyên và tiến trình được coi là một đỉnh. Cạnh là liên kết được tạo từ tài nguyên đến tiến trình được cấp phát hoặc từ tiến trình đến tài nguyên được yêu cầu.
- Trong khoa học máy tính, đồ thị dùng để biểu diễn luồng quy trình tính toán.

Nội dung tiếp theo

**Thuật toán tìm kiếm theo chiều sâu
(DFS)**