Chương 1. Biểu diễn đồ thị

Khi lập trình giải các bài toán được mô hình hoá bằng đồ thị, việc đầu tiên cần làm tìm cấu trúc dữ liệu để biểu diễn đồ thị sao cho việc giải quyết bài toán được thuận tiện nhất.

Có rất nhiều phương pháp biểu diễn đồ thị, trong bài này chúng ta sẽ khảo sát một số phương pháp phổ biến nhất. Tính hiệu quả của từng phương pháp biểu diễn sẽ được chỉ rõ hơn trong từng thuật toán cụ thể.

1.1. Ma trận kề

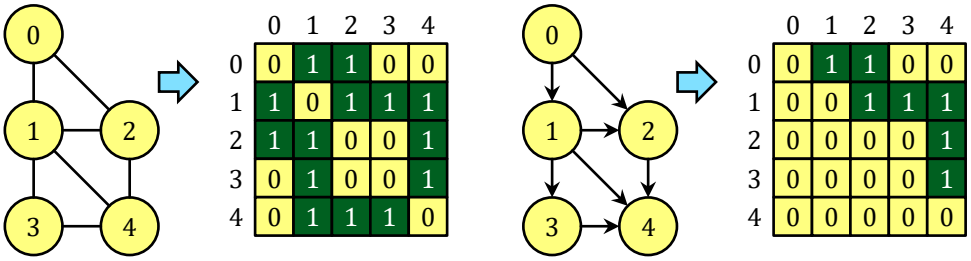
Với $G = (V, E)$ là một đơn đồ thị có hướng trong đó $|V| = n$, ta có thể đánh số các đỉnh từ 0 tới $n - 1$ và đồng nhất mỗi đỉnh với số thứ tự của nó. Bằng cách đánh số như vậy, đồ thị G có thể biểu diễn bằng ma trận vuông $A = \{a_{ij}\}_{n \times n}$ với các hàng và các cột đánh số từ 0 tới $n - 1$. Trong đó:

$$a_{ij} = \begin{cases} 1, \text{nếu } (i,j) \in E \\ 0, \text{nếu } (i,j) \notin E \end{cases}$$

(Cũng có thể dùng hai giá trị kiểu bool: true/false thay cho hai giá trị 0/1)

Với $\forall i$, giá trị của các phần tử trên đường chéo chính ma trận A: $\{a_{ii}\}$ có thể đặt tuỳ theo mục đích cụ thể, chẳng hạn đặt bằng 0. Ma trận A xây dựng như vậy được gọi là *ma trận kề* (*adjacency matrix*) của đồ thị G. Việc biểu diễn đồ thị vô hướng được quy về việc biểu diễn phiên bản có hướng tương ứng: thay mỗi cạnh (i, j) bởi hai cung ngược hướng nhau: (i, j) và (j, i).

Đối với đa đồ thị thì việc biểu diễn cũng tương tự trên, chỉ có điều nếu như (i, j) là cung thì a_{ij} là số cạnh nối giữa đỉnh i và đỉnh j.



Hình 1-1. Ma trận kề biểu diễn đồ thị

Ma trận kề có một số tính chất:

- Đối với đồ thị vô hướng G, thì ma trận kề tương ứng là ma trận đối xứng, $A = A^T$ $(\forall i, j: a_{ij} = a_{ji})$, điều này không đúng với đồ thị có hướng.
- Nếu G là đồ thị vô hướng và A là ma trận kề tương ứng thì trên ma trận A, tổng các số trên hàng i bằng tổng các số trên cột i và bằng bậc của đỉnh i: deg (i)
- Nếu G là đồ thị có hướng và A là ma trận kề tương ứng thì trên ma trận A, tổng các số trên hàng i bằng bậc ra của đỉnh i: $\deg^+(i)$, tổng các số trên cột i bằng bậc vào của đỉnh i: $\deg^-(i)$

Ưu điểm của ma trận kề:

- Đơn giản, trực quan, dễ cài đặt trên máy tính
- Để kiểm tra xem hai đỉnh (u, v) của đồ thị có kề nhau hay không, ta chỉ việc kiểm tra bằng một phép so sánh: $a_{uv} \neq 0$

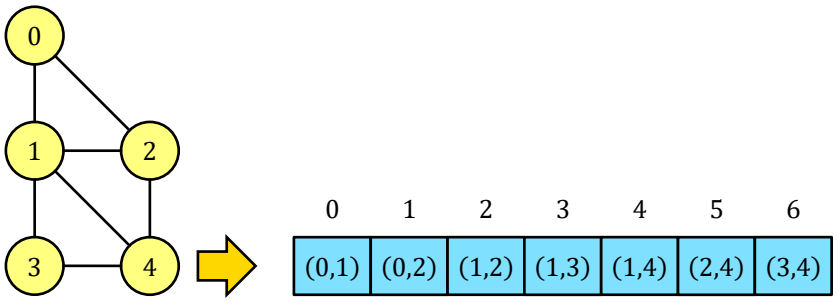
Nhược điểm của ma trận kề

- Bất kể số cạnh của đồ thị là nhiều hay ít, ma trận kề luôn luôn đòi hỏi n^2 ô nhớ để lưu các phần tử ma trận, điều đó gây lãng phí bộ nhớ.
- Một số bài toán yêu cầu thao tác liệt kê tất cả các đỉnh v kề với một đỉnh u cho trước. Trên ma trận kề việc này được thực hiện bằng cách xét tất cả các đỉnh v và kiểm tra điều kiện $a_{uv} \neq 0$. Như vậy, ngay cả khi đỉnh u là *đỉnh cô lập* (không kề với đỉnh nào) hoặc *đỉnh treo* (chỉ kề với 1 đỉnh) ta cũng buộc phải xét tất cả các đỉnh v và kiểm tra giá trị tương ứng a_{uv}.

1.2. Danh sách cạnh

Với đồ thị $G = (V, E)$ có n đỉnh, m cạnh, ta có thể liệt kê tất cả các cạnh của đồ thị trong một danh sách, mỗi phần tử của danh sách là một cặp (x, y) tương ứng với một cạnh của E, trong trường hợp đồ thị có hướng thì mỗi cặp (x, y) tương ứng với một cung, x là đỉnh đầu và y là đỉnh cuối của cung. Cách biểu diễn này gọi là *danh sách cạnh* (*edge list*).

Có nhiều cách xây dựng cấu trúc dữ liệu để biểu diễn danh sách, nhưng phổ biến nhất là dùng mảng hoặc danh sách móc nối.



Hình 1-2. Danh sách cạnh

Ưu điểm của danh sách cạnh:

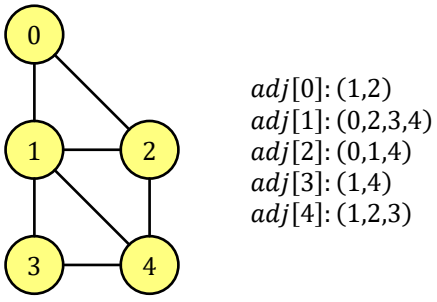
- Trong trường hợp đồ thị thưa (có số cạnh tương đối nhỏ), cách biểu diễn bằng danh sách cạnh sẽ tiết kiệm được không gian lưu trữ, bởi nó chỉ cần $O(m)$ ô nhớ để lưu danh sách cạnh.
- Trong một số trường hợp, ta phải xét tất cả các cạnh của đồ thị thì cài đặt trên danh sách cạnh làm cho việc duyệt các cạnh dễ dàng hơn (Chẳng hạn như thuật toán Kruskal).

Nhược điểm của danh sách cạnh:

- Nhược điểm cơ bản của danh sách cạnh là khi ta cần duyệt tất cả các đỉnh kề với đỉnh v nào đó của đồ thị, thì chẳng có cách nào khác là phải duyệt tất cả các cạnh, lọc ra những cạnh có chứa đỉnh v và xét đỉnh còn lại.
- Việc kiểm tra hai đỉnh u, v có kề nhau hay không cũng bắt buộc phải duyệt danh sách cạnh, điều đó khá tốn thời gian trong trường hợp đồ thị dày (nhiều cạnh).

1.3. Danh sách kề

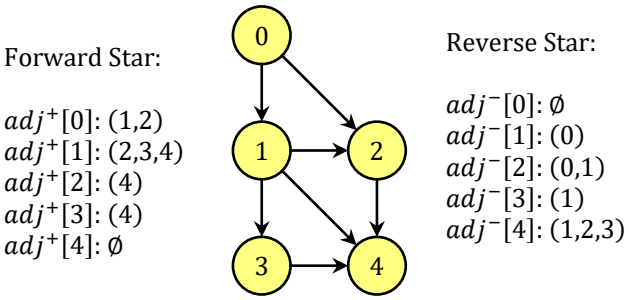
Để khắc phục nhược điểm của các phương pháp ma trận kề và danh sách cạnh, người ta đề xuất phương pháp biểu diễn đồ thị bằng *danh sách kề* (*adjacency list*). Trong cách biểu diễn này, với mỗi đỉnh u của đồ thị vô hướng, ta cho tương ứng với nó một danh sách $adj[u]$ gồm các đỉnh kề với u (Hình 1-4).



Hình 1-3. Danh sách kề

Với đồ thị có hướng $G = (V, E)$. Có hai cách cài đặt danh sách kề phổ biến:

- Forward Star: Với mỗi đỉnh u, lưu trữ một danh sách $adj^+[u]$ chứa các đỉnh nối từ u: $adj^+[u] = \{v: (u,v) \in E\}$.
- Reverse Star: Với mỗi đỉnh v, lưu trữ một danh sách $adj^-[v]$ chứa các đỉnh nối tới v: $adj^-[v] = \{u: (u,v) \in E\}$



Hình 1-4. Danh sách kề forward star và reverse star

Tùy theo từng bài toán, chúng ta sẽ chọn cấu trúc Forward Star hoặc Reverse Star để biểu diễn đồ thị. Có những bài toán yêu cầu phải biểu diễn đồ thị bằng cả hai cấu trúc Forward Star và Reverse Star.

Bất cứ cấu trúc dữ liệu nào có khả năng biểu diễn danh sách (mảng, danh sách móc nối, cây…) đều có thể sử dụng để biểu diễn danh sách kề, nhưng mảng và danh sách móc nối được sử dụng phổ biến nhất. Chẳng hạn trong C++, ta có thể dùng vector, list, set, … để biểu diễn mỗi danh sách $adj[.]$.

Ưu điểm của danh sách kề: Danh sách kề cho phép dễ dàng duyệt tất cả các đỉnh kề với một đỉnh u cho trước.

Nhược điểm của danh sách kề: Danh sách kề yếu hơn ma trận kề ở việc kiểm tra (u,v) có phải là cạnh hay không, bởi trong cách biểu diễn này ta sẽ phải việc phải duyệt toàn bộ danh sách kề của u hay danh sách kề của v.

1.4. Danh sách liên thuộc

Danh sách liên thuộc (incidence lists) là một mở rộng của danh sách kề. Nếu như trong biểu diễn danh sách kề, mỗi đỉnh được cho tương ứng với một danh sách các đỉnh kề thì trong biểu diễn danh sách liên thuộc, mỗi đỉnh được cho tương ứng với một danh sách các cạnh liên thuộc. Chính vì vậy, những kỹ thuật cài đặt danh sách kề có thể sửa đổi một chút để cài đặt danh sách liên thuộc.

1.5. Chuyển đổi giữa các cách biểu diễn đồ thị

Có một số thuật toán mà tính hiệu quả của nó phụ thuộc rất nhiều vào cách thức biểu diễn đồ thị, do đó khi bắt tay vào giải quyết một bài toán đồ thị, chúng ta phải tìm cấu trúc dữ liệu phù hợp để biểu diễn đồ thị sao cho hợp lý nhất. Nếu đồ thị đầu vào được cho bởi một cách biểu diễn bất hợp lý, chúng ta cần chuyển đổi cách biểu diễn khác để thuận tiện trong việc triển khai thuật toán.

Với các lớp mẫu của C++ biểu diễn cấu trúc dữ liệu mảng động, danh sách móc nối, tập hợp, việc chuyển đổi giữa các cách biểu diễn đồ thị khá dễ dàng, ta sẽ trình bày chúng trong chương trình cài đặt các thuật toán. Cũng có một số thuật toán không phụ thuộc nhiều vào cách biểu diễn đồ thị, trong trường hợp này ta sẽ chọn cấu trúc dữ liệu dễ cài đặt nhất để việc đọc hiểu thuật toán/chương trình được thuận tiện hơn.

Trong những chương trình mô tả thuật toán của phần này, nếu không có ghi chú thêm, ta giả thiết rằng các đồ thị được cho có n đỉnh và m cạnh. Các đỉnh được đánh số từ 0 tới $n-1$ và các cạnh đánh số từ 0 tới $m-1$. Các đỉnh cũng như các cạnh được đồng nhất với số hiệu của chúng.

Bài tập 1-1

Cho một đồ thị có hướng n đỉnh, m cạnh được biểu diễn bằng danh sách kề, trong đó mỗi đỉnh u sẽ được cho tương ứng với một danh sách các đỉnh nối từ u. Cho một đỉnh v, hãy tìm thuật toán tính bán bậc ra và bán bậc vào của v. Xác định độ phức tạp tính toán của thuật toán

Bài tập 1-2

Đồ thị chuyển vị của đồ thị có hướng$G = (V, E)$ là đồ thị $G^T = (V, E^T)$, trong đó:

$$E^T = \{(u, v): (v, u) \in E\}$$

Hãy tìm thuật toán xây dựng G^T từ G trong hai trường hợp: G và G^T được biểu diễn bằng ma trận kề; G và G^T được biểu diễn bằng danh sách kề.

Bài tập 1-3

Cho đa đồ thị vô hướng $G = (V, E)$ được biểu diễn bằng danh sách kề, hãy tìm thuật toán $O(|V| + |E|)$ để xây dựng đơn đồ thị $G' = (V, E')$ và biểu diễn G' bằng danh sách kề, biết rằng đồ thị G' gồm tất cả các đỉnh của đồ thị G và các cạnh song song trên G được thay thế bằng duy nhất một cạnh trong G'.

Bài tập 1-4

Cho đa đồ thị G được biểu diễn bằng ma trận kề $A = \{a_{ij}\}$ trong đó a_{ij} là số cạnh nối từ đỉnh i tới đỉnh j. Hãy chứng minh rằng nếu $B = A^k$ thì b_{ij} là số đường đi từ đỉnh i tới đỉnh j qua đúng k cạnh.

Gợi ý: Sử dụng chứng minh quy nạp.

Bài tập 1-5

Cho đơn đồ thị $G = (V, E)$, ta gọi bình phương của một đồ thị G là đơn đồ thị

$$G^2 = (V, E^2)$$

sao cho $(u, v) \in E^2$ nếu và chỉ nếu tồn tại một đỉnh $w \in V$ sao cho (u, w) và (w, v) đều thuộc E.

Hãy tìm thuật toán $O(|V|^3)$ để xây dựng G^2 từ G trong trường hợp cả G và G^2 được biểu diễn bằng ma trận kề, tìm thuật toán $O(|E||V| + |V|^2)$ để xây dựng G^2 từ G trong trường hợp cả G và G^2 được biểu diễn bằng danh sách kề.

Bài tập 1-6

Xây dựng cấu trúc dữ liệu để biểu diễn đơn đồ thị vô hướng và các thao tác:

- Liệt kê các đỉnh kề với một đỉnh u cho trước trong thời gian $O(\deg(u))$
- Kiểm tra hai đỉnh có kề nhau hay không trong thời gian $O(1)$
- Loại bỏ một cạnh trong thời gian $O(1)$
- Bổ sung một cạnh (u, v) nếu nó chưa có trong thời gian $O(1)$

Bài tập 1-7

Với đồ thị $G = (V, E)$ được biểu diễn bằng ma trận kề, đa số các thuật toán trên đồ thị sẽ có độ phức tạp tính toán $\Omega(|V|^2)$, tuy nhiên không phải không có ngoại lệ. Chẳng hạn bài toán tìm "bồn chứa" (*universal sink*) trong đồ thị: bồn chứa trong đồ thị có hướng là một đỉnh nối từ tất cả các đỉnh khác và không có cung đi ra. Hãy tìm thuật toán $O(|V|)$ để xác định sự tồn tại và chỉ ra bồn chứa trong đồ thị có hướng.

Bài tập 1-8

Xét đồ thị vô hướng $G = (V, E)$ với các đỉnh đánh số từ 0 tới $n - 1$ và các cạnh đánh số từ 0 tới $m - 1$. Xây dựng *Ma trận liên thuộc (incidence matrix)* $B = \{b_{ij}\}_{n \times m}$ trong đó:

$$b_{ij} = \begin{cases} 1, \text{ nếu cung } j \text{ liên thuộc với đỉnh } i \\ 0, \text{ trong trường hợp ngược lại} \end{cases}$$

Xét ma trận $A = B.B^T = \{a_{ij}\}_{n \times n}$, chứng minh rằng nếu đồ thị không có khuyên thì:

- $\forall i \neq j$, a_{ij} là số cạnh nối giữa đỉnh i và đỉnh j trên G
- $\forall i$, a_{ii} là bậc của đỉnh i

Nói cách khác, $B.B^T$ là ma trận kề của G, các phần tử trên đường chéo chính tương ứng với bậc đỉnh.

Bài tập 1-9

Xét đồ thị vô hướng $G = (V, E)$ với ma trận liên thuộc B định nghĩa như trong Bài tập 1-8. Xét đồ thị đường $L(G)$, trong đó mỗi đỉnh của $L(G)$ ứng với một cạnh của G. Hai đỉnh i, j của $L(G)$ kề nhau nếu tồn tại một đỉnh u của G liên thuộc với cả hai cạnh i, j. Gọi A là ma trận kề của đồ thị $L(G)$ xác định bởi

$$a_{ij} = |\{u \in G: \text{cả cạnh } i \text{ và cạnh } j \text{ liên thuộc với } u\}|$$

Chứng minh rằng nếu đồ thị G không có khuyên thì $A = B^T.B$