Báo cáo bài tập Phân tích thiết kế giải thuật

**Đề bài:** Bài toán cây bao trùm tối thiểu và thuật toán Kruskal

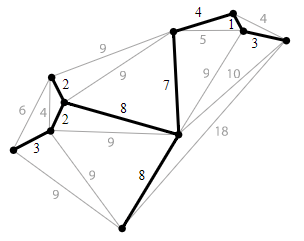
Họ và tên: Phạm Ngọc Dũng

Lớp: Tin học 50

1. **Mô tả bài toán**

Với một đồ thị liên thông, vô hướng cho trước, cây bao trùm của nó là một đồ thị con có dạng cây và có tất cả các đỉnh liên thông với nhau. Một đồ thị có thể có nhiều cây bao trùm khác nhau. Chúng ta gán một trọng số cho mỗi cạnh. Khi đó một **cây bao trùm nhỏ nhất** là một cây bao trùm có tổng trọng số trên các cạnh bé hơn hoặc bằng tổng trọng số của tất cả các cây bao trùm khác.

Tổng quát hơn, bất kỳ một đồ thi vô hướng nào (không nhất thiết phải liên thông) đều có một **rừng bao phủ nhỏ nhất**, là hội của các cây bao trùm nhỏ nhất của các thành phần liên thông của nó.

Hình 1: Hình ảnh thể hiện một cây bao trùm nhỏ nhất trong một đồ thị.

1. **Thuật toán Kruskal**
2. Tư tưởng thuật toán

Thuật toán Kruskal dựa trên mô hình xây dựng cây bao trùm bằng thuật toán hợp nhất.

* Luôn duy trì một rừng các cây đã thăm
* Mỗi bước lựa chọn một cạnh nhỏ nhất để gộp hai cây lại thành một cây

Kruskal cần một sử dụng cấu trúc dữ liệu **union-find** để hỗ trợ. Cấu trúc dữ liệu cần đảm bảo các yêu cầu sau:

makeset(u) : tạo một tập {u} trong O(1)

find(u) : trả lại tập chứa đỉnh u trong O(1)

union(u,v) : hợp 2 tập chứa đỉnh u và v trong O(1)

1. Mô tả thuật toán

Giả sử ta cần tìm cây bao trùm nhỏ nhất của đồ thị G. Thuật toán bao gồm các bước sau:

* Khởi tạo rừng F (tập hợp các cây), trong đó mỗi đỉnh của G tao thành một cây riêng biệt.
* Khởi tạo tập S chứa tất cả các cạnh của G
* Chừng nào S còn khác rỗng và F gồm hơn một cây
* Xóa cạnh nhỏ nhất trong S
* Nếu cạnh đó nối hai cây khác nhau trong F thì them nó vào F và hợp nhất hai cây kề với nó làm một.
* Nếu không thì loại bỏ cạnh đó

Kỹ thuật đánh nhãn đỉnh:

* Ngay tại bước 1 của thuật toán, ta gắn đỉnh I của đồ thj một nhãn la i
* Trong bước 2:
  + Nếu hai đầu cạnh e có cùng nhãn (tức là nhãn của e.v1 và nhãn của e.v2 bằng nhau) thì T+{e} sẽ tạo chu trình, ta không đưa e vào T
  + Ngược lại thì ta đưa e vào T và thực hiện công việc ghép nhãn bằng cách:
    - Label1 = Min(Label(e.v1), Label(e.v2))
    - Label2 = Max(Label(e.v1), Label(e.v2))
    - Sửa nhãn của tất cả các đỉnh nào có nhãn là Label2 thành nhãn Label1

1. **Đánh giá độ phức tạp thuật toán**

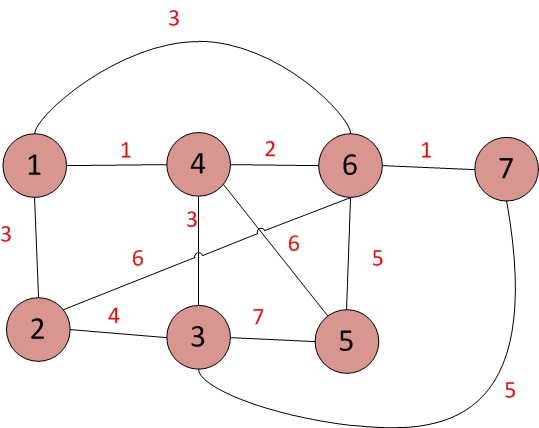
Nếu E là số cạnh và V là số đỉnh của đồ thị thì thuật toán Kruskal chạy trong thời gian O(E\*log(V))

Có thể đạt được thời gian này bằng phương pháp sau: sắp xếp tất cả các cạnh theo trọng số trong thời gian O(E\*log(E)). Điều này cho phép thực hiện “xóa cạnh nhỏ nhất trong S” trong thời gian O(1). Sau đó sử dụng cấu trúc dữ liệu union-find để lưu trữ thông tin đỉnh nào nằm ở tập nào. Ta cần thực hiện O(E) thao, hai thao tác tìm và một thao tác hợp cho mỗi cạnh. Như vậy tổng thời gian là O(E\*log(E)) = O(E\*log(V))

1. **Ví dụ**

Cho đồ thị G như hình vẽ: Yêu cầu tìm cây khung nhỏ nhất của đồ thị G.

G gồm có 7 đỉnh và 12 cạnh. Thuật toán sẽ dừng khi có 6 cạnh trong tập T



**Bước 1:** Khởi tạo T = φ, D(T) = 0

**Bước 2:** Sắp xếp các cạnh theo chiều không giảm trọng số

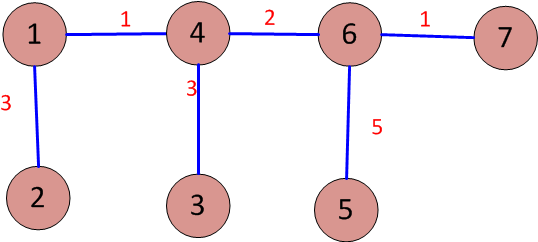
|  |  |  |
| --- | --- | --- |
| Điểm đầu | Điểm cuối | Trọng số |
| 1 | 4 | 1 |
| 6 | 7 | 1 |
| 4 | 6 | 2 |
| 1 | 2 | 3 |
| 1 | 6 | 3 |
| 3 | 4 | 3 |
| 2 | 3 | 4 |
| 3 | 7 | 5 |
| 5 | 6 | 5 |
| 2 | 6 | 6 |
| 4 | 5 | 6 |
| 3 | 5 | 7 |

**Bước 3:** Lặp

|  |  |  |
| --- | --- | --- |
| STT | Cạnh được xét | T ∪ e |
| 1 | E\(1,4) | T ∪ (1,4); D(T) = 1 |
| 2 | E\(6,7) | T = T ∪ (6,7); D(T) = 1 + 1 = 2 |
| 3 | E\(4,6) | T = T ∪ (4,6); D(T) = 2 + 2 = 4 |
| 4 | E\(1,2) | T = T ∪ (1,2); D(T) = 4 + 3 = 7 |
| 5 | E\(1,6) | Tạo nên chu trình |
| 6 | E\(3,4) | T = T ∪ (3,4); D(T) = 7 + 3 = 10 |
| 7 | E\(2,3) | Tạo nên chu trình |
| 8 | E\(3,7) | Tạo nên chu trình |
| 9 | E\(5,6) | T = T ∪ (5,6); D(T) = 10 + 5 = 15 |

Thuật toán dừng lại khi ta đã tìm được 6 cạnh.

**Kết quả:** Ta được đồ thị sau



Với tổng trọng số trên đồ thị là 15