**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**🕯✡🕮🕮✡🕯**



**BÁO CÁO ĐỒ ÁN CÔNG NGHỆ THÔNG TIN**

**TUẦN 11**

***ĐỀ TÀI:* GRAPH**

**GVHD**: ThS. Nguyễn Quang Ngọc

**Sinh viên thực hiện**:

|  |  |
| --- | --- |
| **HỌ VÀ TÊN** | **MÃ SỐ SINH VIÊN** |
| Nguyễn Minh Quang | 20143481 |
| Đỗ Minh Cường | 21110147 |

## Tiến độ nhiệm vụ

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tiến trình hoàn thành môn Đồ án CNTT | | | | | | | | | | | | | | | | | | |
|  | Các phương thức | TKB | | | | | | | | | | | | | Cường | | Quang | |
| 1 | Tạo 1 đồ thị | o | o |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |
| 2 | Thêm một đỉnh vào đồ thị đã có |  |  | o |  |  |  |  |  |  |  |  |  |  | |  | |  | |
| 3 | Thay đổi thông tin của một đỉnh |  |  | o |  |  |  |  |  |  |  |  |  |  | |  | |  | |
| 4 | Thêm cạnh |  |  |  | o |  |  |  |  |  |  |  |  |  | |  | |  | |
| 5 | Thay đổi trọng số của cạnh |  |  |  |  | o |  |  |  |  |  |  |  |  | |  | |  | |
| 6 | Xuất các tên đỉnh, tên cạnh |  |  |  | o |  |  |  |  |  |  |  |  |  | |  | |  | |
| 7 | Xuất thông tin 1 đỉnh, 1 cạnh | o | o |  |  | o |  |  |  |  |  |  |  |  | |  | |  | |
| 8 | Xuất ma trận kề, ma trận liên kết |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |
| 9 | Duyệt đồ thị theo chiều rộng, chiều xâu |  |  |  |  |  |  | o |  |  |  |  |  |  | |  | |  | |
| 10 | Đường đi ngắn nhất từ đỉnh v đến đỉnh w |  |  |  |  |  | o |  |  |  |  |  |  |  | |  | |  | |
| 11 | Kiểm tra có chu trình Euler |  |  |  |  |  |  |  | o |  |  |  |  |  | |  | |  | |
| 12 | Cây khung bé nhất |  |  |  |  |  |  |  |  | o |  |  |  |  | |  | |  | |
| 13 | Tìm dòng chảy lớn nhất (NetworkModel) |  |  |  |  |  |  |  |  |  | o |  |  |  | |  | |  | |
| Day | | 03/09/2023 | 10/09/2023 | 17/09/2023 | 24/09/2023 | 24/09/2023 | 30/09/2023 | 7/10/2023 | 14/10/2023 | 21/10/2023 | 28/10/2023 | 4/11/2023 |  |  | |  | |  | |
| Week | | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 |  |  | |  | |  | |
| Note | |  |  |  |  |  |  |  |  |  | o – Begin  o – Complete 50%  o – Complete 100% | | | | | | | | |

## So sánh giữa 2 thuật toán đánh nhãn và thuật toán Ford-Fulkerson:

Thuật toán đánh nhãn có một số tương đồng với thuật toán Ford-Fulkerson, tuy nhiên, nó không hoàn toàn giống với Ford-Fulkerson và có một số khác biệt quan trọng.

Có một số điểm chính khác biệt giữa thuật toán này và Ford-Fulkerson:

* Duyệt đỉnh chưa được duyệt với chỉ số i bé nhất: Trong Ford-Fulkerson, không có yêu cầu về việc duyệt đỉnh chưa được duyệt với chỉ số i bé nhất. Ford-Fulkerson thường duyệt theo đỉnh nào có cạnh trong mạng luồng còn khả dụng để tăng cường dòng chảy.
* Sử dụng nhãn (α, Δ) cho đỉnh và các cạnh: Ford-Fulkerson thường sử dụng giá trị còn dư trên cạnh để quyết định việc tăng cường dòng chảy, trong khi thuật toán bạn mô tả sử dụng các nhãn (α, Δ) để quản lý việc duyệt và điều kiện dừng.
* Gán nhãn dựa trên điểm α, Δ và dòng chảy trên cạnh: Thuật toán này gán nhãn cho các đỉnh và cạnh dựa trên các điểm α và Δ, cùng với thông tin về dòng chảy trên các cạnh. Ford-Fulkerson thường tập trung vào cộng hoặc giảm dòng chảy trên các cạnh dựa trên giá trị còn dư.
* Độ phức tạp: Cả hai thuật toán có độ phức tạp về thời gian không xác định trước do phụ thuộc vào số lần lặp và kích thước mạng.
* Đường tăng luồng: Cả hai thuật toán đều tìm đường tăng luồng để tăng luồng trong mạng.
* Tính hiệu quả: Thuật toán Ford-Fulkerson có thể bị kẹt ở một vòng lặp vô hạn nếu không áp dụng một chiến lược lựa chọn cụ thể cho đường tăng luồng. Trong khi đó, thuật toán đánh nhãn sử dụng nhãn để tối ưu hóa việc tìm đường tăng luồng và tránh việc kẹt ở vòng lặp vô hạn.
* Lát cắt cực tiểu: Cả hai thuật toán đều xây dựng lát cắt cực tiểu của mạng sau khi tìm được luồng cực đại.
* Nhược điểm của thuật toán Ford-Fulkerson là cần một chiến lược lựa chọn đường tăng luồng cụ thể để đảm bảo kết thúc trong thời gian hợp lý. Thuật toán đánh nhãn giải quyết vấn đề này bằng cách sử dụng nhãn để tối ưu hóa việc tìm đường tăng luồng, giúp tránh việc kẹt ở vòng lặp vô hạn.
* Cách quản lý đỉnh và cạnh: Trong thuật toán đánh nhãn, bạn sử dụng các nhãn (α, Δ) để quản lý các đỉnh và cạnh, trong khi trong Ford-Fulkerson, thường sử dụng giá trị còn dư trên các cạnh để tăng cường dòng chảy.
* Thứ tự duyệt đỉnh: Thuật toán đánh nhãn sử dụng một cách tiếp cận đặc biệt, yêu cầu duyệt đỉnh chưa được duyệt với chỉ số i bé nhất. Trong khi đó, Ford-Fulkerson tập trung vào duyệt các đỉnh mà có cạnh còn dư trong mạng luồng.
* Phương pháp xác định đường đi tăng cường: Edmonds-Karp, một biến thể của Ford-Fulkerson, sử dụng tìm đường đi ngắn nhất bằng thuật toán BFS để tăng cường dòng chảy. Trong khi đó, thuật toán đánh nhãn sử dụng nhãn (α, Δ) để xác định đường đi tăng cường.
* Hiệu suất thời gian: Thuật toán Edmonds-Karp có thời gian chạy tối ưu hơn trong nhiều trường hợp do việc sử dụng BFS để tìm đường đi tăng cường. Thuật toán đánh nhãn có thể có thời gian chạy tốt hơn trong một số trường hợp cụ thể, nhưng nó không thường được ưa chuộng như Edmonds-Karp.
* Khả năng tránh kẹt mạng: Edmonds-Karp sử dụng BFS để tìm đường đi tăng cường, điều này giúp tránh kẹt mạng. Thuật toán đánh nhãn có khả năng gây kẹt mạng nếu không được thực hiện cẩn thận.