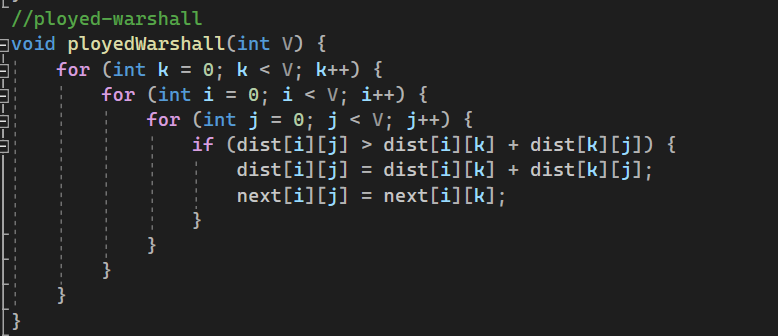
THUẬT TOÁN PLOYED WARSHALL

Thuật toán Floyd-Warshall là một thuật toán quan trọng trong lý thuyết đồ thị, được đặt theo tên của hai nhà khoa học đã đóng góp vào việc phát triển nó, Robert Floyd và Stephen Warshall. Thuật toán này được sử dụng để tìm khoảng cách ngắn nhất giữa mọi cặp đỉnh trong một đồ thị có hướng hoặc không hướng, có trọng số hoặc không trọng số.

Dưới đây là mã giả cho thuật toán Floyd-Warshall và giải thích về các hàm trong chương trình:

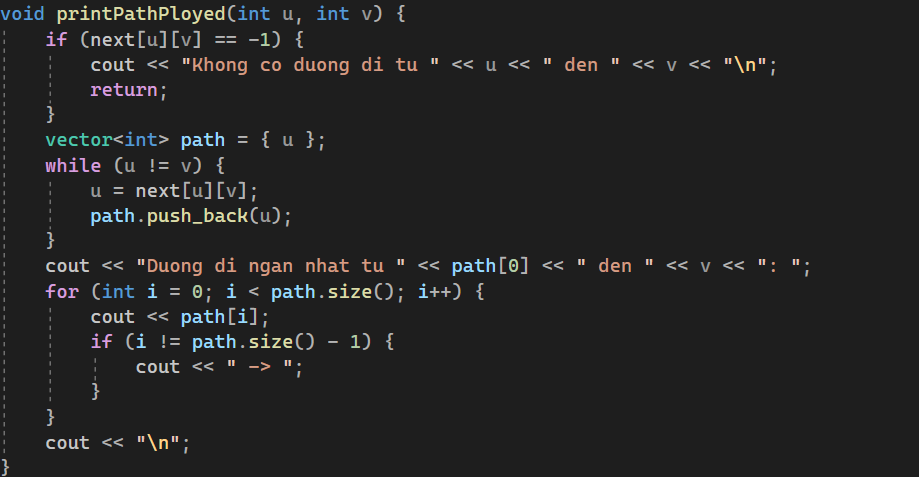
***Hàm ployedWarshall(V):***

******

1. Duyệt qua mọi cặp đỉnh (i, j) với k là đỉnh trung gian.

2. Nếu dist[i][j] > dist[i][k] + dist[k][j] thì cập nhật dist[i][j] = dist[i][k] + dist[k][j] và next[i][j] = next[i][k].

***Hàm printPathPloyed(u, v):***

******

1. Nếu không có đường đi từ u đến v (next[u][v] == -1), in ra thông báo và thoát hàm.

2. Khởi tạo một vector để lưu đường đi.

3. Trong khi u khác v, thêm u vào đường đi và cập nhật u = next[u][v].

4. In ra đường đi từ đỉnh u đến v.

***Hàm nhapDoThiTrongSo(a, n, m):***

1. Mở tệp "inputDijkstra.txt".

2. Đọc số lượng đỉnh n và cạnh m.

3. Đọc danh sách các cạnh và trọng số tương ứng, cập nhật ma trận a.

4. Đóng tệp.

```

Giải thích các hàm:

- `ployedWarshall`: Hàm này thực hiện thuật toán Floyd-Warshall để tính toán khoảng cách ngắn nhất giữa mọi cặp đỉnh trong đồ thị.

- `printPathPloyed`: Hàm này in ra đường đi ngắn nhất từ đỉnh u đến đỉnh v. Nếu không có đường đi, hàm sẽ in ra thông báo không có đường đi.

- `nhapDoThiTrongSo`: Hàm này đọc dữ liệu đầu vào từ tệp "inputDijkstra.txt", bao gồm số lượng đỉnh, số lượng cạnh và danh sách các cạnh cùng với trọng số tương ứng.

Trình bày thuật toán:

Thuật toán Floyd-Warshall bắt đầu bằng việc khởi tạo ma trận khoảng cách `dist` và ma trận `next`. Ma trận `dist` lưu khoảng cách ngắn nhất giữa mọi cặp đỉnh, trong khi ma trận `next` lưu đỉnh kế tiếp trên đường đi ngắn nhất từ một đỉnh đến một đỉnh khác.

Sau đó, thuật toán sẽ duyệt qua mọi cặp đỉnh `(i, j)` với `k` là đỉnh trung gian. Nếu khoảng cách từ `i` đến `j` thông qua `k` ngắn hơn khoảng cách hiện tại từ `i` đến `j`, thuật toán sẽ cập nhật khoảng cách và đỉnh kế tiếp.

Cuối cùng, thuật toán sẽ in ra đường đi ngắn nhất từ đỉnh bắt đầu đến đỉnh kết thúc. Nếu không có đường đi giữa hai đỉnh, thuật toán sẽ thông báo không có đường đi. Chú ý rằng, bạn cần nhập cặp đỉnh bắt đầu và kết thúc sau khi nhập danh sách các cạnh.