TRƯỜNG ĐẠI HỌC SƯ PHẠM THÀNH PHỐ HỒ CHÍ MINH

**KHOA CÔNG NGHỆ THÔNG TIN**

🙢⬩🙠



**BÁO CÁO ĐỀ TÀI**

**XÂY DỰNG CHƯƠNG TRÌNH GIẢI BÀI TOÁN DIJSKTRA TRỰC QUAN**

***Giảng viên hướng dẫn:*** TS. Nguyễn Đình Hiển

***Nhóm sinh viên thực hiện (Nhóm Decade):***

**1.** Phùng Tiến Đạt 44.01.104.074

**2.** Ngụy Thế Dương 44.01.104.072

**3.** Võ Minh Quốc Việt 44.01.104.234

Tháng 3/2019

**LỜI CẢM ƠN**

Để thực hiện và hoàn thành đề tài này, nhóm chúng em đã nhận được sự hỗ trợ, giúp đỡ cũng như là quan tâm từ nhiều phía. Đề tài cũng được hoàn thành dựa trên sự tham khảo, học tập kinh nghiệm từ các kết quả của nhiều tác giả. Đặc biệt hơn nữa là sự giúp đỡ, tạo điều kiện về vật chất và tinh thần từ phía gia đình, bạn bè và các đồng nghiệp.

Trước hết, em xin gửi lời cảm ơn sâu sắc đến Thầy **Nguyễn Đình Hiển** – người trực tiếp hướng dẫn nhóm chúng em trong suốt quá trình học tập và hoàn thành đề tài.

Tuy có nhiều cố gắng, nhưng trong đề tài này không tránh khỏi những thiếu sót. Em kính mong các thầy tiếp tục có những ý kiến đóng góp, giúp đỡ để đề tài được hoàn thiện hơn.

Một lần nữa nhóm chúng em xin chân thành cám ơn!

**ĐỀ TÀI – PHÂN CÔNG**

|  |  |  |
| --- | --- | --- |
| Tên đề tài | Xây dựng chương trình giải bài toán Dijskstra trực quan | |
| Lĩnh vực chuyên ngành | Trí tuệ nhân tạo | |
| Loại hình thực hiện | Sản phẩm | |
| Người hướng dẫn | TS. Nguyễn Đình Hiển | |
| Tên nhóm | Decade | |
| Sinh viên thực hiện 1 | Phùng Tiến Đạt | 44.01.104.074 |
| Sinh viên thực hiện 2 | Ngụy Thế Dương | 44.01.104.072 |
| Sinh viên thực hiện 3 | Võ Minh Quốc Việt | 44.01.104.234 |

🖎 **Phân công công việc:**

|  |  |  |
| --- | --- | --- |
| TT | Họ và tên | Phân công |
| **1** | Phùng Tiến Đạt | Tìm tài liệu và code  Tiến độ: Hoàn thành trước **14/11/2020**. |
| **2** | Ngụy Thế Dương | Tìm tài liệu và thiết kế giao diện  Tiến độ: Hoàn thành trước **14/11/2020**. |
| **3** | Võ Minh Quốc Việt | Tìm tài liệu và hoàn chỉnh báo cáo  Tiến độ: Hoàn thành trước **14/11/2020**. |

**MỤC TIÊU – NỘI DUNG ĐỀ TÀI – PHƯƠNG PHÁP**

Với nội dung được phân công trong học phần Trí tuệ nhân tạo, áp dụng kiến thức đã học để giải quyết bài toán tìm đường đi ngắn nhất và tạo một chương trình có giao diện minh họa cụ thể, nay **nhóm Decade** hoàn thiện **Dijsktra** **Demo** với nhiều mong muốn nhất định.

*Thứ nhất,* toàn bộ chương trình Dijsktra Demo được thiết kế dựa trên thuật toán chính là **Dijsktra** - bài toán đường đi ngắn nhất trong một đồ thị không có cạnh trọng số âm.

*Thứ hai,* thật sự hy vọng chương trình sẽ là một phương tiện cụ thể để sinh viên tiếp cận thuật toán nhanh hơn, dễ hiểu hơn trong học phần Lý thuyết đồ thị.

**Phương pháp thực hiện:**

Sử dụng thư viện đồ họa Winform của C# để tạo giao diện trực quan cho bài toán tìm đường đi ngắn nhất bằng thuật toán Dijskstra.

**GIỚI THIỆU THUẬT TOÁN**

Giải thuật **Dijkstra**, mang tên của một nhà khoa học máy tính người Hà Lan Edsger W. Dijkstra, là một thuật toán giải quyết bài toán **đường đi ngắn nhất** trong một đồ thị có hướng **không có cạnh trọng số âm**. Ứng dụng lớn nhất của thuật toán này là trong công nghệ Hệ thống định vị toàn cầu (GPS).



*Edsger W. Dijkstra – nhà khoa học máy tính người Hà Lan*

**Tóm tắt thuật toán:**

Cho G = (X, E) là một đồ thị có trọng không âm gồm n đỉnh. Thuật toán Dijkstra được dùng để tìm đường đi ngắn nhất từ đỉnh x đến y cho trước.

☞ **Input:** Đồ thị G, đỉnh xuất phát x, đỉnh đích y.

☜ **Output:** Đường đi ngắn nhất từ x đến y.

🖎 **Thuật toán:**

**Bước 1**: Gán T := X

Length[x] = 0; Length[k] =+, kX\{x}

LastV[k] = -1, .

**Bước 2:** Nếu y T thì dừng.

**Bước 3:** Chọn đỉnh v T sao cho Length[v] nhỏ nhất và loại v khỏi T: T := T\{v}.

**Bước 4:** và có cạnh nối từ v đến k,

Nếu Length[k]>Length[v]+Lvk thì

Length[k]=Length[v]+ Lvk

LastV[k]=v

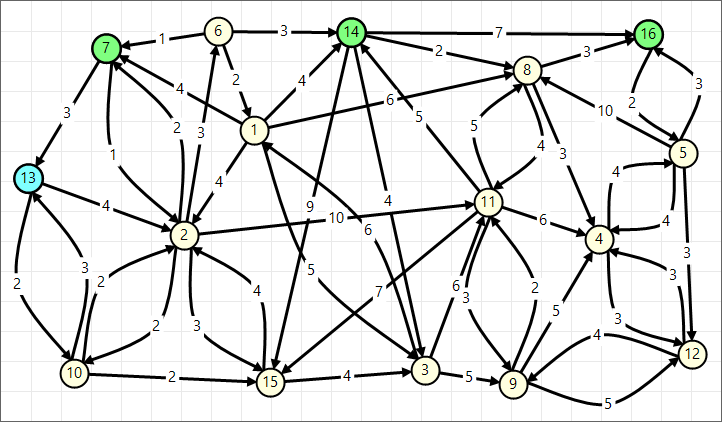
**Bước 5:** Tiếp tục với Bước 2.

**Chú ý:** Khi thuật toán dừng, nếu Length[j] = +∞ thì không tồn tại đường đi từ i đến j, nếu ngược lại thì Length[j] là độ dài đường đi ngắn nhất.

**MÔ TẢ THUẬT TOÁN BẰNG DIJKSTRA DEMO**

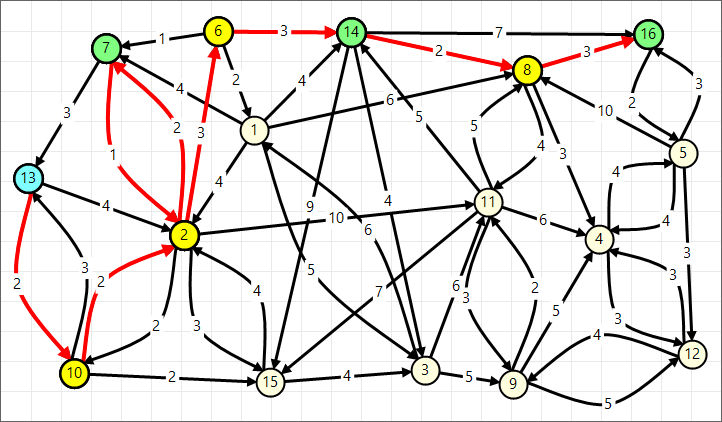
Bài toán được đặt ra ở đây là: Một người giao hàng được nhận địa chỉ của các điểm cần giao hàng tới, tuy nhiên vì mới còn là người mới chưa thực sự rành đường đi hoặc không nhớ đường tốt, người giao hàng sẽ có thể đi một đoạn đường có thể xa, tốn kém để đi giao tới các điểm.

Giả sử ta mô phỏng bản đồ dưới dạng đồ thị và đánh dấu các điểm đi



Ở đây ta sẽ chọn điểm 13 là điểm xuẩt phát, các điểm phải tới là 7 14 16

Chạy Chương trình Dijkstra Demo ta sẽ thu được kết quả:



**Thuật toán** xây dựng lộ trình ở đây là:

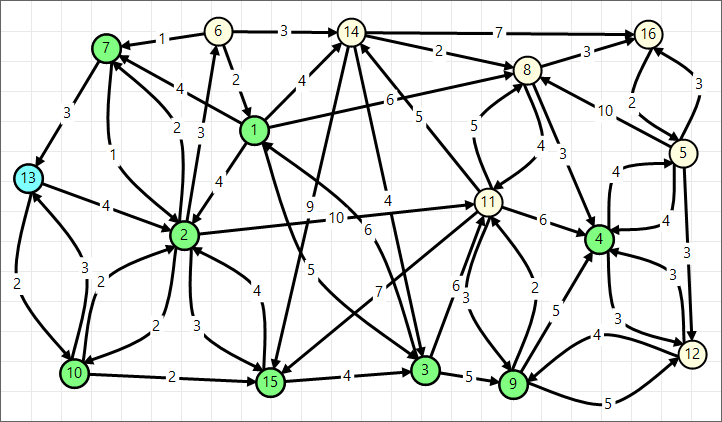
Bước 1: Chọn điểm bắt đầu, được đặt là điểm 13.

Bước 2: Đầu tiên, ta xét điểm 13 đến các đỉnh còn lại, bằng cách chạy Dijkstra cho từng điểm, ta thu được khoảng cách từ đỉnh 13 đến các đỉnh còn lại, ta lấy ra điểm đến ngắn nhất với đỉnh 13, ở đây là 7. Đặt điểm bắt đầu là 7, loại 13 ra khỏi danh sách đỉnh.

Bước 3: Nếu danh sách đỉnh không còn đỉnh ta dừng thuật toán và xuất ra kết quả. Ngược lại ta quay lại bước 1.

Tuy nhiên thuật toán đến đây chưa phải hoàn chỉnh

Ta xét một VD khác là điểm xuất phát là 13 đến các điểm 1 4 7 9 15 10 3 2



Nếu ta chạy theo thuật toán trên, ta có:



Nhưng nếu ta xét chính xác hơn, vẫn còn tồn tại một lộ trình ngắn hơn nữa là:

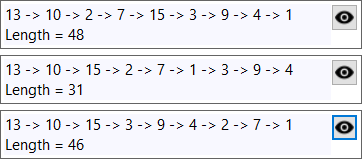


Để ý sự khác biệt thì ta nhận ra rằng trọng số là đỉnh 15 – 3 cũng bằng với đỉnh 15 – 2, vậy để tạo một lộ trình ta sẽ phải xét tất cả các trường hợp để có thể đưa ra lộ trình ngắn nhất có thể

Thuật toán trên sẽ được cập nhật lại ở bước 2:

Nếu tồn tại các đỉnh có trọng số bằng nhau, ta lưu các đỉnh này vào một danh sách tạm, rồi lần lượt xét các đỉnh trong danh sách tạm. Các đỉnh trong danh sách tạm sẽ được bắt đầu lại từ bước 1 đến bước 3.

Đây là kết quả cuối cùng khi chạy thuật toán đã sửa đổi



Ta có thể tưởng tượng nó như một cây phân nhánh.

Thuật toán cũng có nhược điểm của nó là thời gian chạy có thể sẽ khá lâu với một danh sách dữ liệu lớn

**Áp dụng thực tế:** Hiện tại thuật toán mới chỉ chạy trên nền tảng lý thuyết, trên thực tế ta sẽ còn phải tính đến các điều kiện ảnh hưởng đến trọng số của đồ thị như là: Kẹt xe, giờ cao điểm, đường hỏng đột xuất, tốc độ chạy, hết nhiên liệu…

**KẾT QUẢ ĐỀ TÀI**

Sau một thời gian làm việc, thì chương trình cũng hoàn chỉnh theo như mong muốn của các thành viên trong nhóm. Chương trình đáp ứng được các yêu cầu sau:

*Thứ nhất,* xây dựng kết hợp vừa như dụng cụ học tập của sinh viên, vừa giải quyết vấn đề thực tế đặt ra ban đầu.

*Thứ hai,* hoạt động của chương trình cũng tương đối hoàn thiện: các công cụ dễ dùng, khai thác có hiệu quả các tính năng của chương trình.

*Thứ ba,* đầu tư về giao diện rõ ràng, thân thiện với người sử dụng.

*Thứ tư,* việc cài đặt và thử nghiệm sau nhiều lần kiểm tra, sửa lỗi, góp ý và cuối cùng cũng thật sự hài lòng với kết quả cuối cùng.

|  |
| --- |
| **ĐÁNH GIÁ CỦA GIẢNG VIÊN** |
| TS. Nguyễn Đình Hiển |
|  |