TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

BÁO CÁO PROJECT 1
Website chạy thuật toán TSP – Travelling Salesman Problem sử dụng phương pháp Backtracking

Trần Thành Vinh – 20225539
Vinh.TT225539@sis.hust.edu.vn

Ngành Khoa học dữ liệu và Trí tuệ nhân tạo
Lớp DSAI – 03

Giảng viên hướng dẫn: Nguyễn Văn Sơn

Viện/trường: Công nghệ thông tin và truyền thông

HÀ NỘI, 1/2025

Nội dung báo cáo

Báo cáo project 1 này là một website chạy thuật toán Backtracking cho bài toán Travelling Salesman Problem (TSP). Website có thể còn nhiều lỗi trong quá trình chạy do thời gian hoàn thiện gấp rút. Source code sử dụng, bao gồm khung HTML và CSS, cùng với các Animation elements được implement từ David Galles, University of San Francisco.

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU CHUNG

1.1 Lý thuyết

1.1.1 Đồ thị

Đồ thị là một tập hợp bao gồm các đỉnh (nodes) và các cạnh nối giữa các đỉnh (edges) với nhau. Thông thường trong tin học, đồ thị sẽ có 2 loại chính là: có hướng và vô hướng.

Đồ thị vô hướng G là 1 cặp đỉnh không có thứ tự (unordered pair) G := (V,E). Trong đó:

- V là tập các **đỉnh** hoặc các **nút**.
- E là tập chứa cặp đỉnh không có thứ tự, hoặc là **cạnh**.

Đồ thị thường có hai loại, vô hướng (undirected) và có hướng (directed). Trên đồ thị vô hướng thì các cạnh đi từ nút A sang nút B có thể được sử dụng để đi từ nút B sang nút A. Trên đồ thị có hướng thì điều này là không chắc chắn.

Đồ thị có nhiều ứng dụng trong thực tế, cũng như trong các khía cạnh của tin học.

1.1.2 Các bài toán trên đồ thị

1.1.2.1. Bài toán tìm đường đi ngắn nhất (Shortest Path)

Bài toán này được phát biểu: "Tìm một đường đi ngắn nhất giữa hai đỉnh của một đồ thị sao cho trọng số của đường đi đó (tức tổng trọng số của các cạnh tạo nên đường đi đó) là nhỏ nhất". Đây là bài toán tìm đường đi ngắn nhất nguồn đơn (Single pair shortest path).

Bài toán còn tồn tại dưới dạng "Tìm các đường đi ngắn nhất từ một đỉnh đến tất cả các đỉnh còn lại", hay chính là bài toán tìm đường đi ngắn nhất giữa mọi cặp đỉnh (All pairs shortest path).

Một số thuật toán nổi tiếng được sử dụng để giải quyết bài toán này có thể kể đến: **Thuật toán Dijkstra** - giải bài toán nguồn đơn nếu tất cả các trọng số đều không âm. Thuật toán này có thể tính toán tất cả các đường đi ngắn nhất từ một đỉnh xuất phát cho trước s tới mọi đỉnh khác mà không làm tăng thời gian chạy, **Thuật toán Bellman-Ford** - giải bài toán nguồn đơn trong trường hợp trọng số có thể có giá trị âm, và **Thuật toán tìm kiếm A*** - giải bài toán nguồn đơn sử dụng heuristics để tăng tốc độ tìm kiếm.

1.1.2.2. Bài toán tìm chu trình Hamilton (Hamilton cycle)

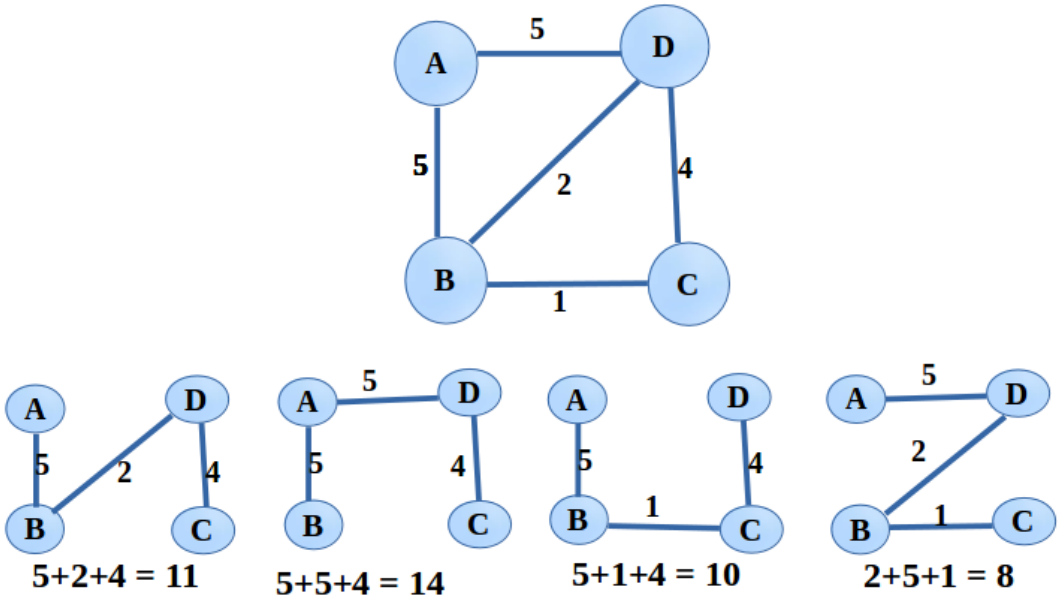
Bài toán này được phát biểu: "Tìm một đường đi sao cho nó đi qua tất cả các nút trong đồ thị đúng 1 lần duy nhất và sau đó quay trở về điểm xuất phát", được gọi theo tên của William Rowan Hamilton.

Thuật toán tổng quát để giải quyết bài toán trên hiện chưa được phát hiện, tuy nhiên, bài toán có thể được giải quyết bằng các phương pháp vét cạn hoặc quay lui và các thuật toán heuristic.

1.1.2.3. Bài toán tìm cây khung nhỏ nhất (Minimum Spanning Tree – MST)

Trước khi nói về bài toán, ta cần hiểu thế nào là một cây? Thế nào là một cây khung? Cây là một đồ thị *liên thông*, tức là tại mọi đỉnh đều tồn tại đường đi đến tất cả các đỉnh trong đồ thị, và không tồn tại chu trình.

Bởi vậy, cây khung được định nghĩa là một cây có đỉnh và cạnh là tập con của tập đỉnh và cạnh của một đồ thị sao cho nó chứa tất cả các đỉnh của đồ thị đó. Một ví dụ của cây khung có thể được nhìn thấy ở ảnh phía dưới.



Hình 1.1 Ví dụ minh họa về cây khung

Như trên hình 1.1, ta có thể thấy các cây khung có trọng số khác nhau. Một đồ thị có thể có nhiều cây khung, do đó bài toán trên mong muốn tìm ra được cây khung có trọng số nhỏ nhất. Hai thuật toán thường được sử dụng để giải quyết bài toán trên là **thuật toán Prim** và **thuật toán Kruskal**.

1.1.2.4. Bài toàn người bán hàng – định tuyến đường đi tối ưu (Travelling Salesman Problem – TSP)

Đây là một bài toán vô cùng kinh điển của tối ưu hóa, xuất phát từ thực tiễn ở các ngành giao thông vận tải, vận chuyển, … . Bài toán này mong muốn sẽ tìm ra được một đường đi đi qua tất cả các nút của một đồ thị và quay về với điểm xuất phát với trọng số nhỏ nhất.

Ta có thể sẽ liên tưởng đến bài toán tìm chu trình Hamilton, với trọng số nhỏ nhất. Tuy nhiên, nên nhớ rằng đối với bài toán TSP thì chỉ có 1 source node duy nhất đó là nhà xưởng, hay là depot.

Bài toán TSP có rất nhiều biến thể, với một biến thể phổ biến là bài toán TSPTW, hay Travelling Salesman Problem with Time Windows. Ở đây, không những phải tìm đường đi tối ưu nhất, mà bài toán còn yêu cầu phải tới các nút trong một khoảng thời gian nhất định. Trong một số trường hợp bài toán này thường gắn với công việc giao hàng, tức sẽ có thêm cả thời gian "giao hàng" tại mỗi nút.

Bài toán TSP hiện nay đã có rất nhiều các giải được nghiên cứu và thử nghiệm, từ những các giải đơn giản như quay lui – backtracking hay nhánh cận – branch and bound, cho đến những thuật toán Constraint programming hay quy hoạch ràng buộc, và các phương pháp heuristic.

Trong project này, bài toán TSP sẽ được giải thông qua phương pháp Backtracking hay là quay lui. Sở dĩ thuật toán trên được lựa chọn là bởi tính iterative (theo lượt) của nó, khiến cho việc trực quan hóa trở nên dễ dàng hơn.

1.1.3 Ứng dụng thực tiễn

Project này có thể được sử dụng như một công cụ giúp người học cấu trúc dữ liệu và giải thuật hoặc tối ưu hóa có thể hình dung được thuật toán một cách dễ dàng hơn.

1.2 Bài toán Travelling Salesman Problem

Trong project, bài toán này sẽ được biểu diễn dưới dạng một đồ thị đơn giản (ít hơn 20 nút). Thuật toán sẽ chạy cho đến khi tìm được đường đi tối ưu nhất theo phương pháp quay lui. Tuy nhiên, bởi vì quay lui yêu cầu tìm kiếm trên toàn bộ không gian tìm kiếm nên thời gian cần thiết để chạy thuật toán là rất lớn.

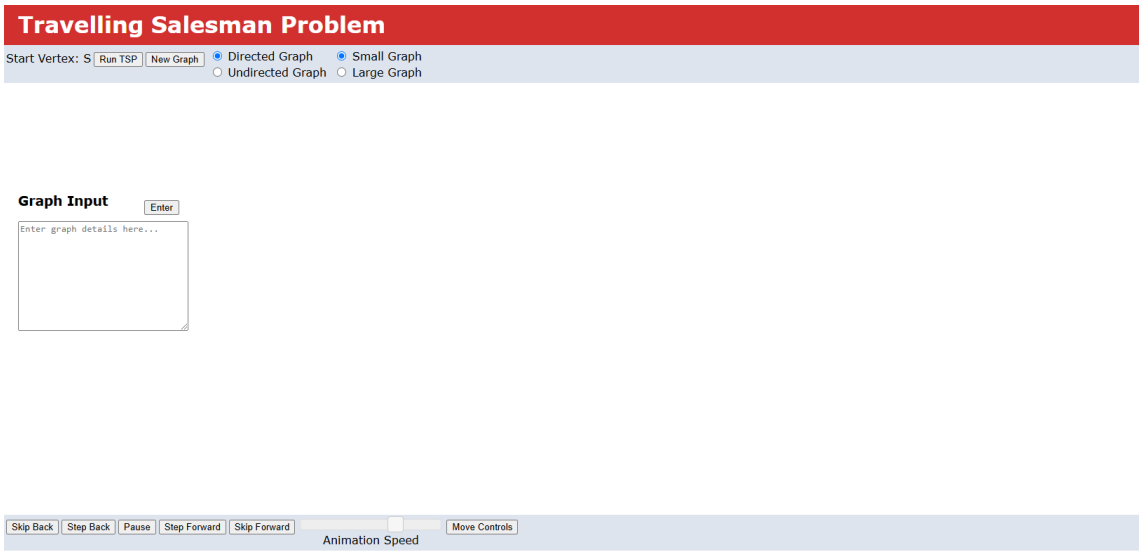
CHƯƠNG 2. CÁC CHỨC NĂNG CHÍNH CỦA TRANG WEB

2.1 Giao diện

Trang web có giao diện giống như hình 2.1, gồm có phần canvas để vẽ đồ thị và phần Graph input để nhập thông tin đồ thị dưới dạng:

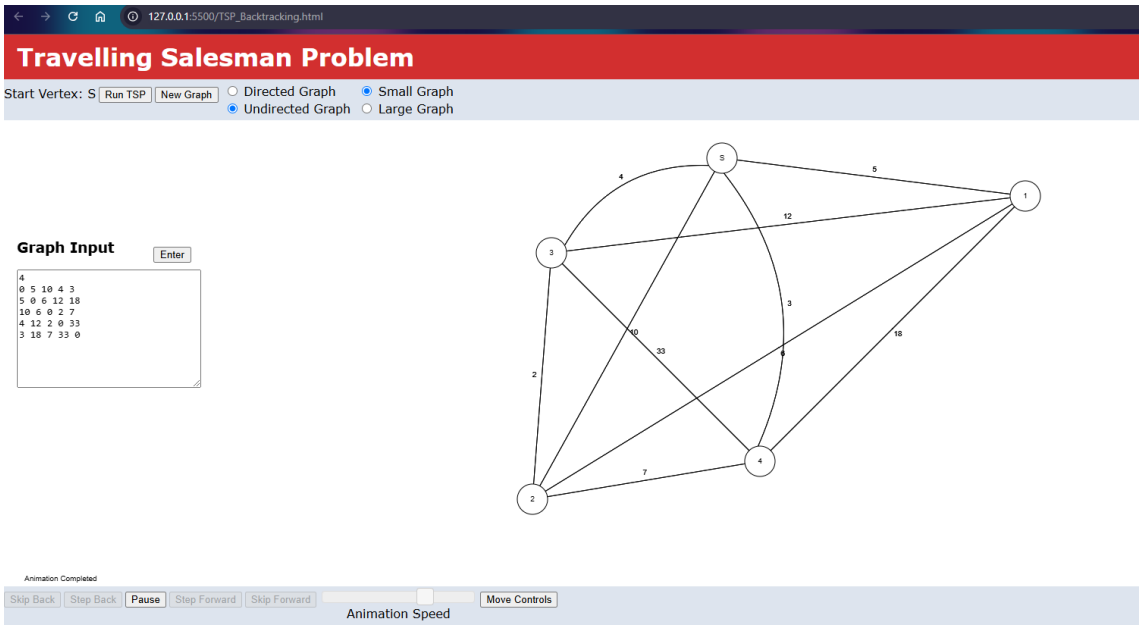
Dòng 1: số các nút (N) khác điểm S (depot)

Dòng 2 - N+2: ma trận kề của đồ thị cần chạy thuật toán backtracking.



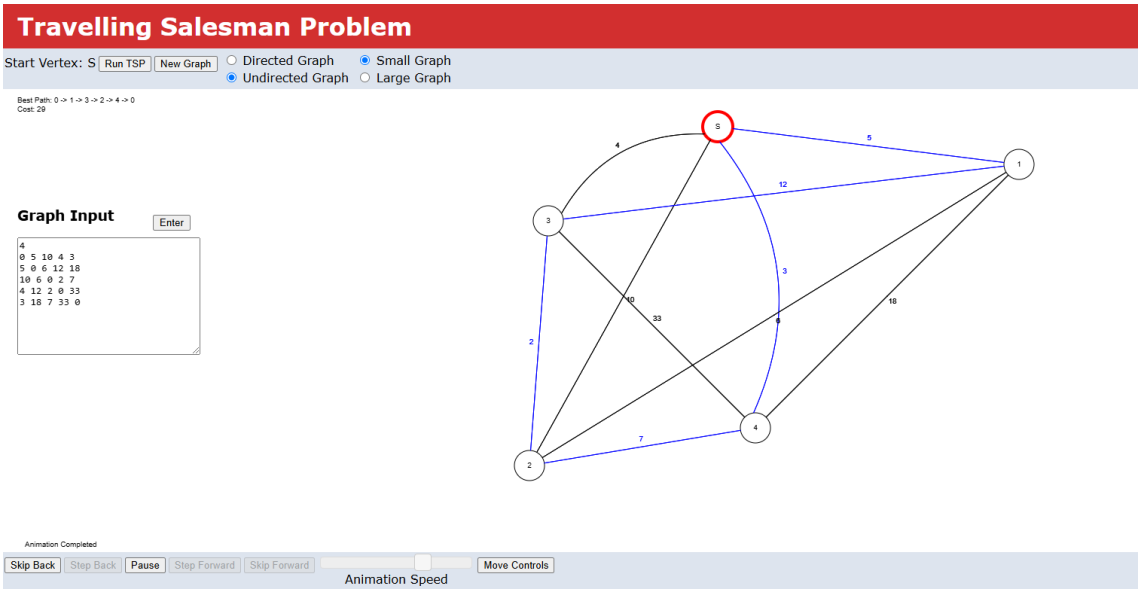
Hình 2.1 Giao diện trang web

Sau khi nhập ma trận kề thì đồ thị sẽ được vẽ như hình 2.2.



Hình 2.2 Ví dụ về đồ thị vô hướng

Sau đó click vào nút “Run TSP”, chờ và thuật toán sẽ chạy đưa ra kết quả trên góc trên cùng bên tay trái (Hình 2.3).

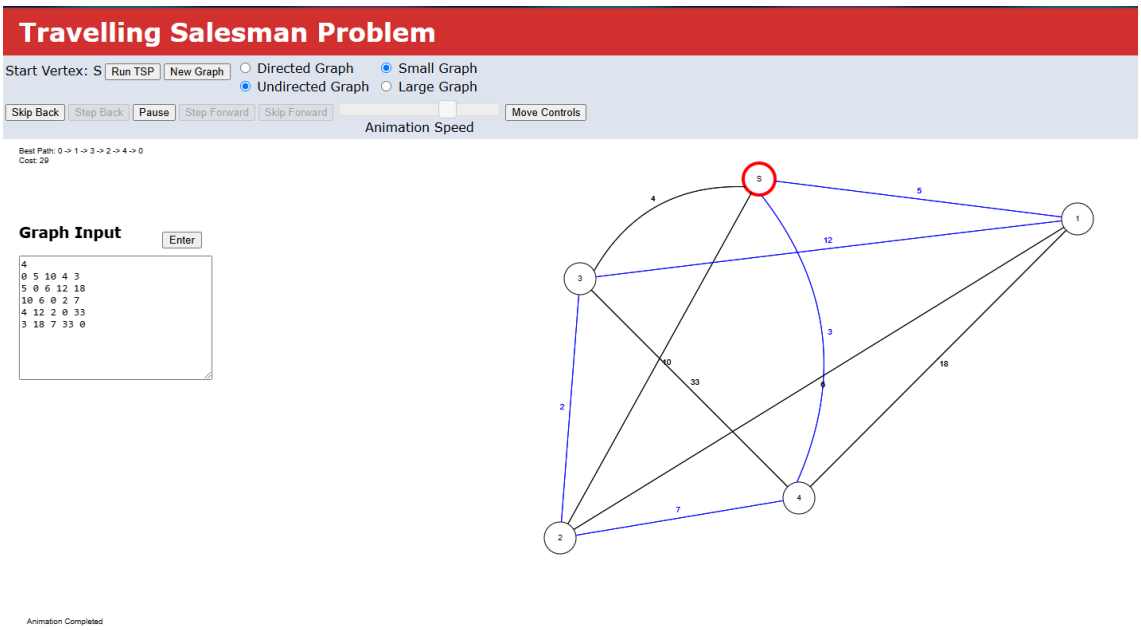


Hình 2.3 Kết quả chạy thuật toán Backtracking cho đồ thị

2.2 Các chức năng

2.2.1 Chức năng Move Control

Chức năng này giúp người dùng di chuyển bảng điều khiển ở phía dưới lên trên cùng, ngay phía dưới của thanh phía trên (Hình 2.4).



Hình 2.4 Kết quả sau khi ấn Move Controls

2.2.2 Chức năng Animation Speed và các tính năng liên quan

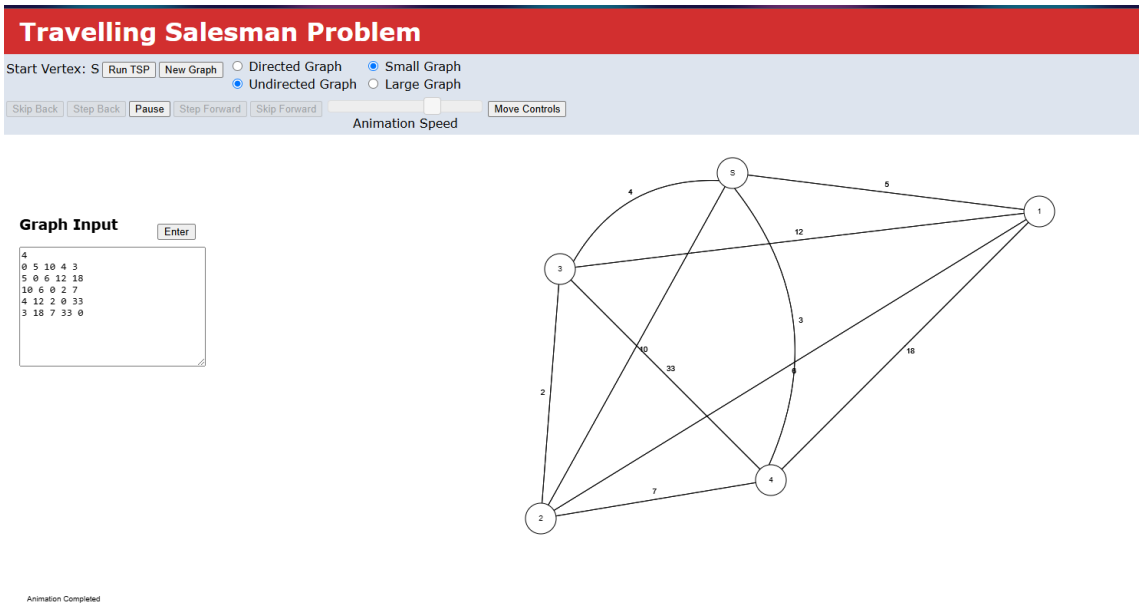
Toàn bộ thành công cụ phía dưới giúp người dùng tăng hoặc giảm tốc độ thuật toán chạy và có thể quay về các bước nếu muốn. Người dùng cũng có thể skip đến cuối (tức là chạy thuật toán như bình thường) và in ra đường đi ngắn nhất trên đồ thị.

2.2.3 Chức năng chọn Small graph Large graph (không khả dụng)

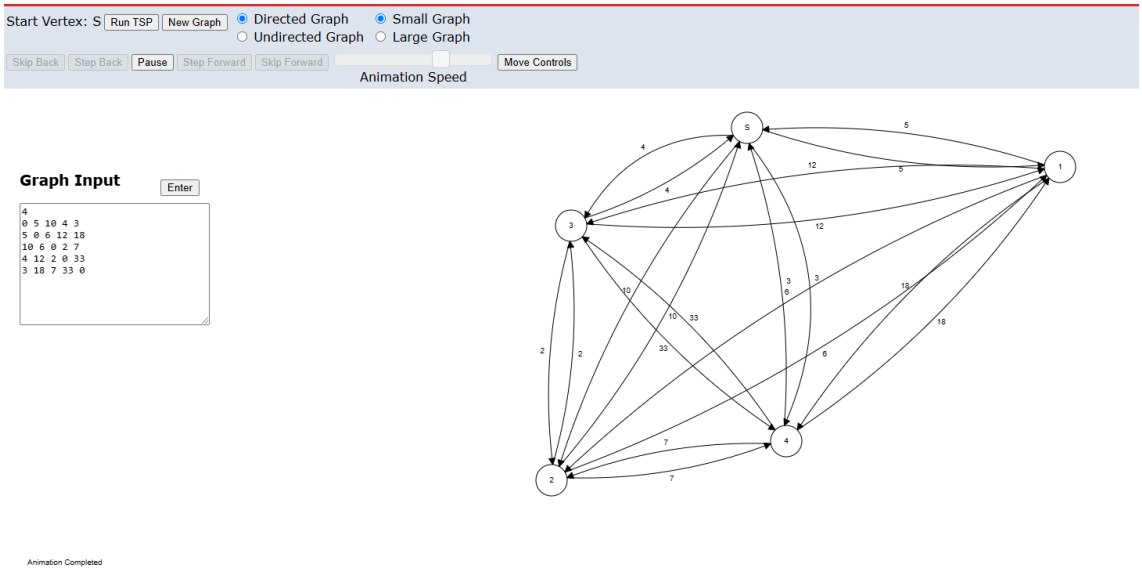
Đây là chức năng giúp người dùng thay đổi một layout tối ưu hơn để vẽ các đồ thị lớn. Tuy nhiên, chức năng này hiện không khả dụng.

2.2.4 Chức năng chọn đồ thị vô hướng hay có hướng

Chức năng này giúp người dùng chọn giữa vẽ đồ thị vô hướng hay có hướng. Nếu người dùng cố tình chọn đồ thị vô hướng, mặc dù ma trận không đối xứng, thì kết quả sẽ bị lỗi. Ví dụ sự khác nhau về đồ thị vô hướng và có hướng. (Hình 2.5 và 2.6)



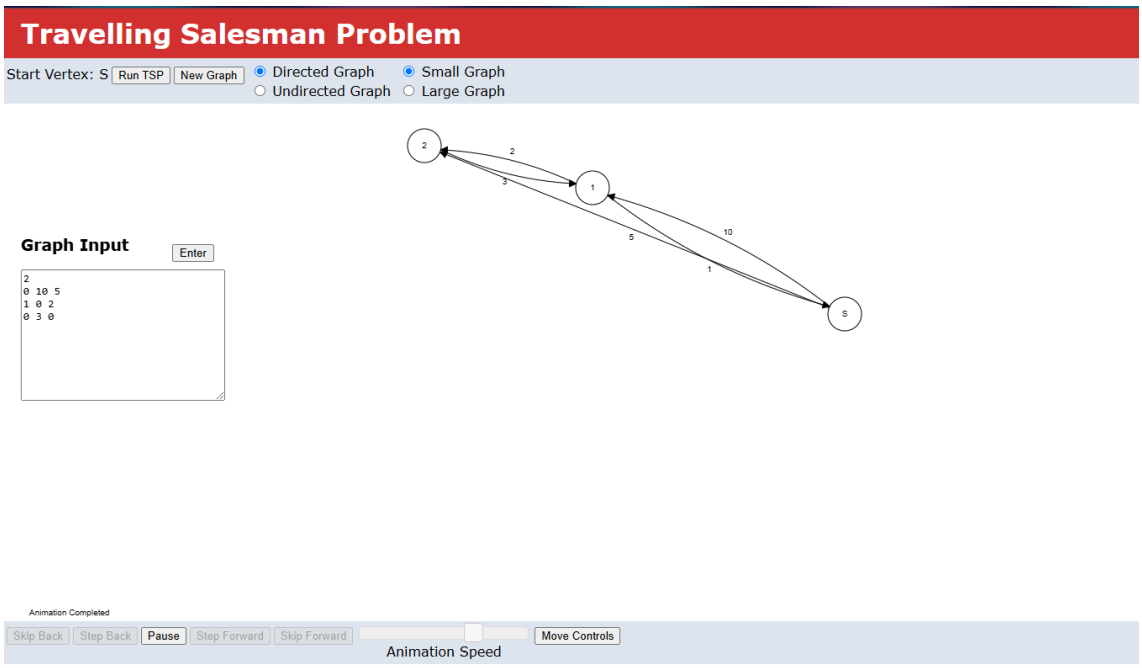
Hình 2.5 Ví dụ về đồ thị vô hướng



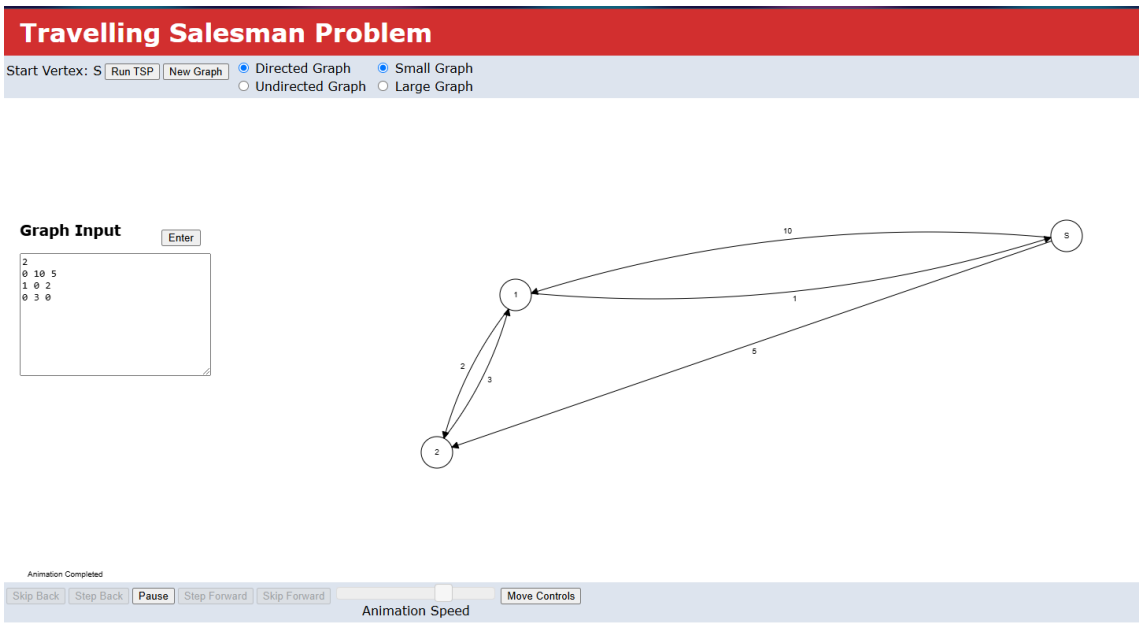
Hình 2.6 Ví dụ về đồ thị có hướng

2.2.5 Chức năng đổi hình đồ thị New Graph

Sau khi người dùng nhập thông tin đồ thị xong, ấn New Graph để đồ thị chuyển thiết kế, nhằm dễ trực quan hơn. (Hình 2.7 và 2.8)



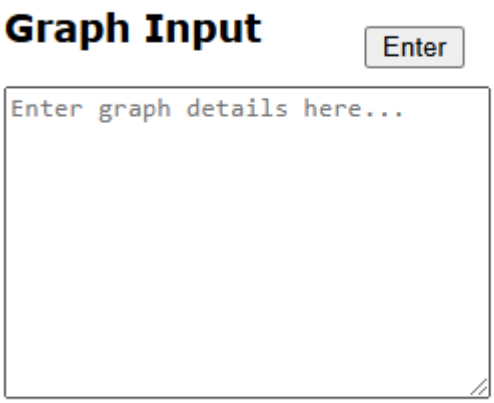
Hình 2.7 Đồ thị sinh ra lần 1



Hình 2.8 Sau khi ấn New graph

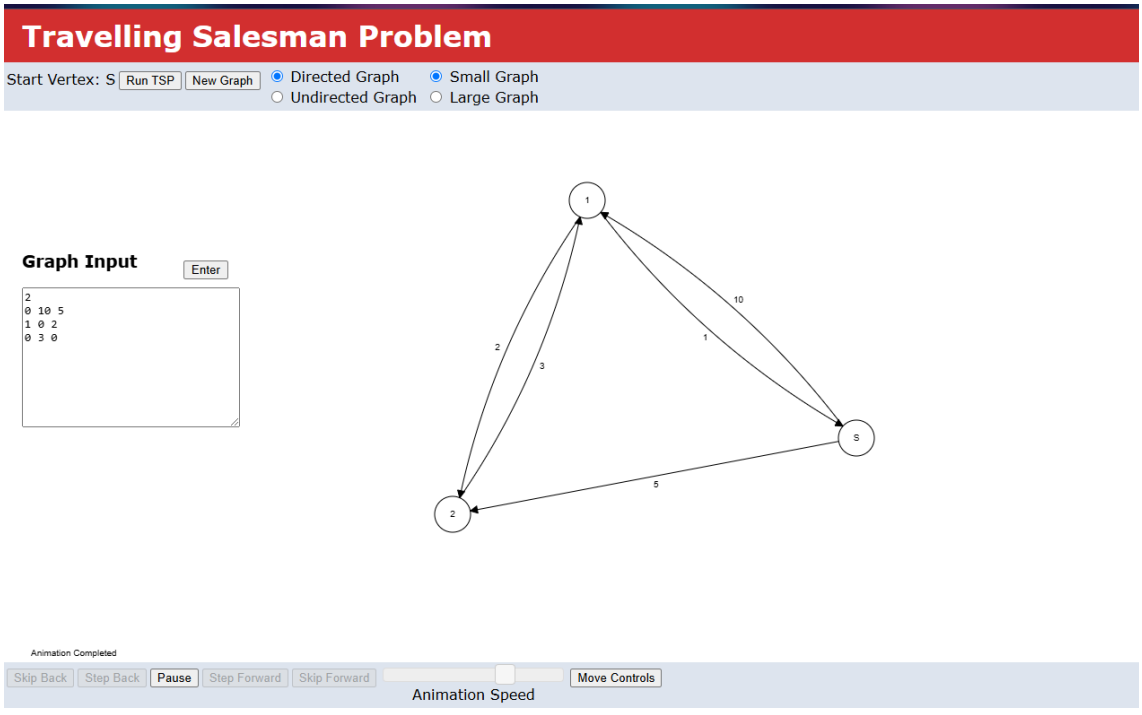
2.3 Demo và hướng dẫn sử dụng

Đầu tiên cần truy cập vào trang web được host local TSP_backtracking.html. Sau đó nhập thông tin đồ thị vào cửa sổ Graph input (Hình 2.9).



Hình 2.9 Cửa sổ Graph input

Sau đó thì ấn Enter để lưu đồ thị, ấn New Graph để vẽ đồ thị mình đã nhập vào. Sau khi hoàn thành các bước trên thì ấn Run TSP để chạy thuật toán. Thuật toán sẽ chạy lần lượt và sẽ dừng khi ở góc màn hình có chữ "Animation Completed". Tuy nhiên đôi khi dòng chữ trên có thể bị in ra mà thuật toán còn chưa chạy. (Hình 2.10)



Hình 2.10 "Animation Completed" nhưng đồ thị vẫn chưa chạy

CHƯƠNG 3. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trang web hoạt động mượt mà với các trường hợp số nút nhỏ hơn hoặc bằng 9, là vừa đủ để trực quan hóa.

Trong tương lai, trang web sẽ hoàn thiện các chức năng còn lại và phát triển thêm các thuật toán khác.