

HOW TO USE PROGRAM

The screenshot shows a software window titled "Floyd-Warshall's Algorithm". It features a tabbed interface with "Enter Matrix" and "Advance" tabs. The "Enter Matrix" tab contains three input areas: "Enter Matrix" (top left), "Information" (top right), and "List_Edge" (bottom left). Below these are "Reset" and "Submit" buttons. The "Advance" tab is currently selected, showing a "Picture Board" area. At the bottom, there are tabs for "Floyd-Warshall" and "Kruskal", a "From:" and "To:" dropdown menu, a "Start" button, and a "See More" section with a link to "Kruskal In Delaunay Triangulation". A footer note reads: "Input Matrix, Submit And Then Draw Vertices In The Picture To The Right".

Thực Thi hàm Floyd-Warshall.

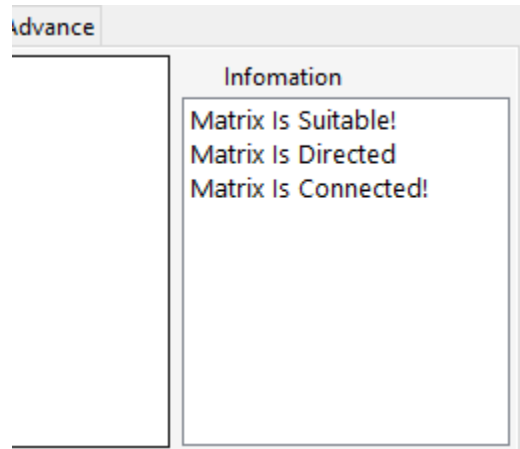
Đầu tiên nhập input là 1 ma trận vào bảng input phía trên bên trái.

Lưu ý: Input là ma trận không cần đỉnh, cuối đỉnh ma trận không được xuống dòng, để đảm bảo nhập đúng.

The screenshot shows the "Enter Matrix" input field with the following 7x7 matrix:

0	3	0	7	0	0	0
8	0	2	0	2	0	0
5	0	0	1	0	0	0
2	0	0	0	0	0	0
0	2	0	0	0	2	0
0	0	0	0	0	0	2
0	0	0	0	0	0	0

Sau khi nhập một ma trận, nhấn nút **SUBMIT** để input dữ liệu vào, Và chương trình sẽ xử lý và kiểm tra ma trận. Kết quả sẽ hiện trên bảng Information kế bên.



Sau khi kiểm tra thành công, thực hiện bước đánh dấu các đỉnh trên Picture Board các đỉnh đầy đủ trong đồ thị, sau khi đánh dấu tất cả các thông tin các cạnh sẽ hiện trong bảng List_Edges.

Enter Matrix **Advance**

Information
 Matrix Is Suitable!
 Matrix Is Directed
 Matrix Is Connected!

List_Edge

- 2--> 1: 8
- 3--> 1: 5
- 4--> 1: 2
- 1--> 2: 3
- 5--> 2: 2
- 2--> 3: 2
- 1--> 4: 7
- 3--> 4: 1

Picture Board

Floyd-Warshall's Algorithm

From:

To:

Result:

Checking Complete, Ready For Submit!

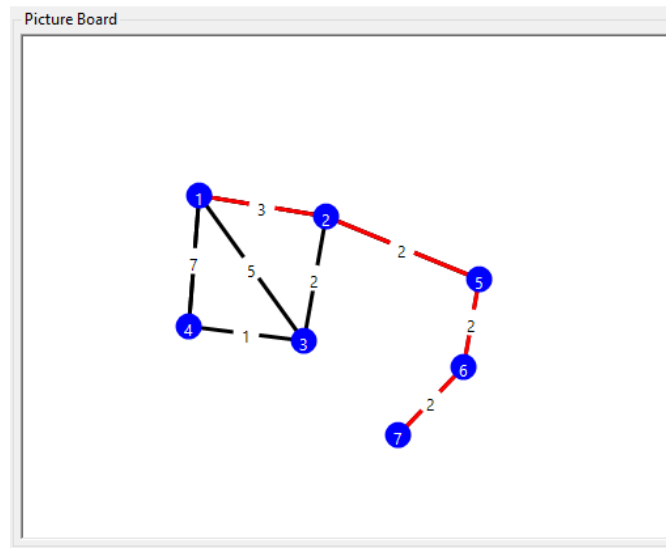
Nhập đoạn đường từ đỉnh đầu (From) với đỉnh đích (To).

Floyd-Warshall's Algorithm

From:

To:

Nhấn nút Start để bắt đầu thuật toán Floyd-Warshall để tìm con đường ngắn nhất của 2 đỉnh.



Ta sẽ thấy từ Picture Board sẽ highlight đoạn đường từ 1—>7 (From→To) là con đường ngắn nhất. Trên hộp text sẽ hiện kết con đường ngắn nhất và trọng số của nó.

Result:

1--->2--->5--->6--->7

Distance is : 9

Nâng cao: Ta có thể thấy được bảng giá trị và bảng hiện các đường đến đỉnh đi của nó, (-1 là con không có con đường từ dòng đến cột).

Enter Matrix Advance

Cost Table

0	3	5	6	5	7	9
5	0	2	3	2	4	6
3	6	0	1	8	10	12
2	5	7	0	7	9	11
7	2	4	5	0	2	4
-1	-1	-1	-1	-1	0	2
-1	-1	-1	-1	-1	-1	0

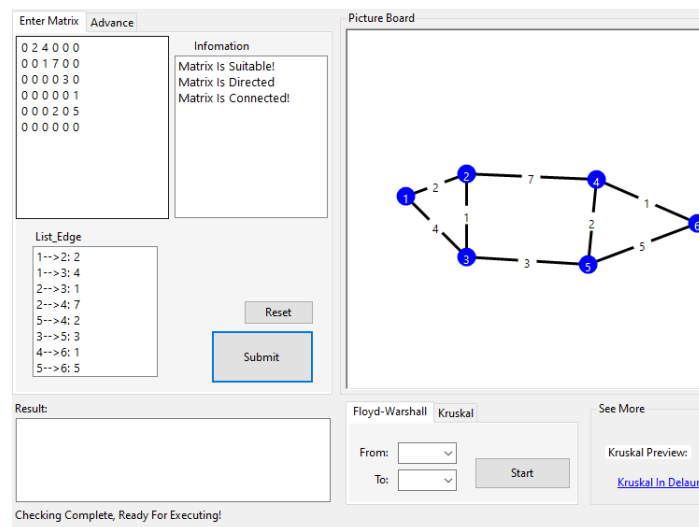
Path Table

0	0	1	2	1	4	5
3	1	1	2	1	4	5
3	0	2	2	1	4	5
3	0	1	3	1	4	5
3	4	1	2	4	4	5
5	5	5	5	5	5	5
6	6	6	6	6	6	6

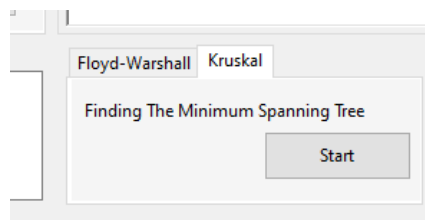
Thực Thi thuật toán Kruskal (Minimum Spanning Tree)

Sau khi một ma trận được nhập và được nhấn **Submit** và đã kiểm tra.

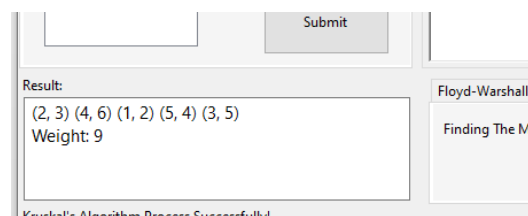
Lưu ý: Khi ma trận đã được nhập và đã thực thi bởi thuật toán Floyd-Warshall, không cần phải nhập lại ma trận mà vẫn được dùng lại ma trận đó luân phiên (Bảng Picture vẫn sẽ hiện theo kết quả khác nhau)



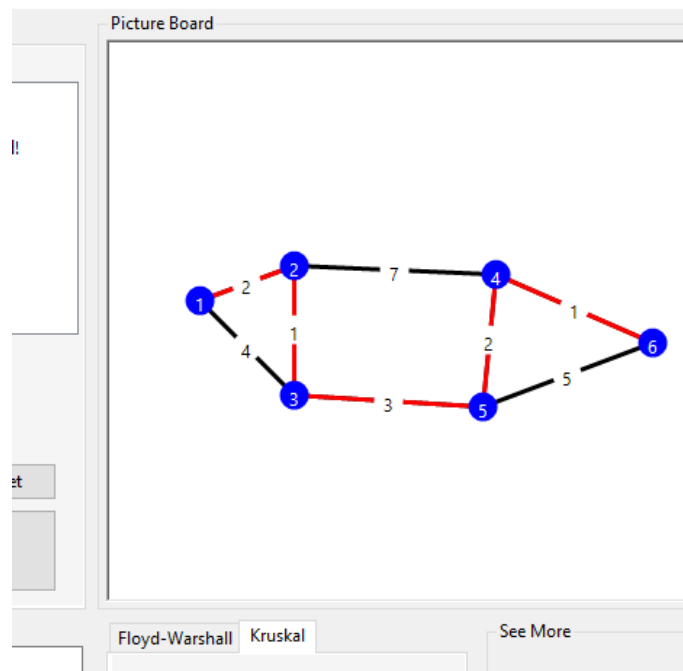
Từ thanh Tab bên dưới (Floyd-Warshall) ta nhấn qua tab thứ hai (Kruskal).



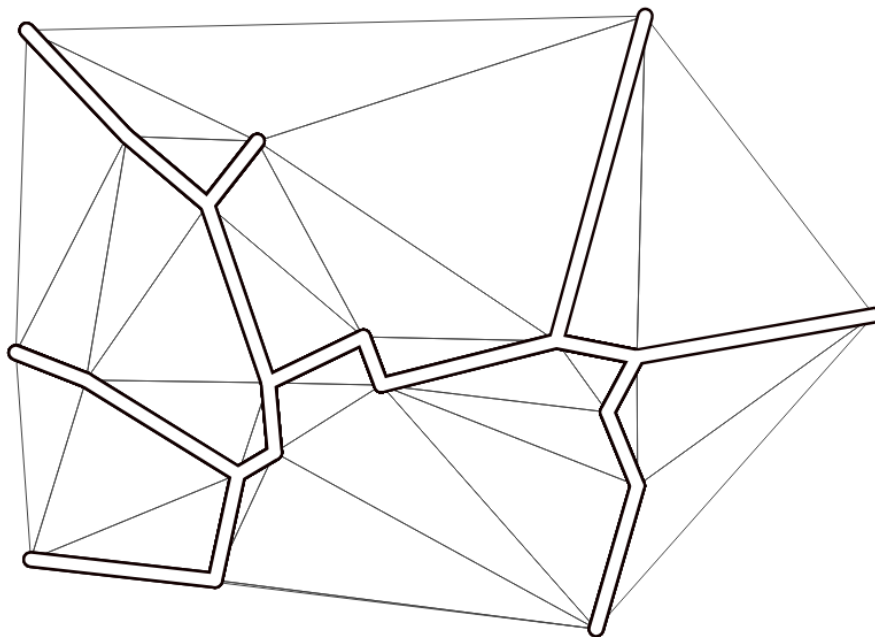
Nhấn nút **Start** để bắt đầu thực thi thuật toán Kruskal, kết quả là các cạnh sẽ được ghi ra ở Result.



Từ bảng PictureBox ta thấy được cây khung với trọng số là ngắn nhất, được vẽ bằng màu đỏ là kết quả (Nếu các đỉnh chưa được đánh dấu, ta vẫn có thể đánh dấu lại đầy đủ các đỉnh).



See More: Thuật toán tìm cây khung nhỏ nhất trong mạng lưới tam giác ngẫu nhiên ***Delaunay Triangulation***.



Source: “***KruskalInDelaunayTriangulation.html***” (Javascript)

(cùng đính kèm trong thư mục project).