TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN**

**TOÁN TỔ HỢP VÀ ĐỒ THỊ**

*Người hướng dẫn*: **GV. LÊ ĐÌNH THẬN**

*Người thực hiện*: **NGUYỄN TỰ THANH DUY – 51503219**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2019**

LỜI CẢM ƠN

Theo sự phân công của Khoa Công Nghệ Thông Tin Trường Đại Học Tôn Đức Thắng, em đã thực hiện đề tài này. Nhờ sự hướng dẫn, giảng dạy nhiệt tình của các thầy(cô) nên em đã có thêm kiến thức để em có thể hoàn thành được đề tài này.Em xin chân thành cảm ơn các thầy cô trong khoa và đặc biệt thầy Lê Đình Thận, người đã trực tiếp hướng dẫn và truyền đạt cho em rất nhiều kiến thức quý báo trong suốt thời gian học tập tại trường, nếu không có thầy em đã không thể một mình hoàn thành được đề tài này. Tuy vậy nhưng kiến thức của em còn hạn chế và còn nhiều bỡ ngỡ nên không thể tránh khỏi những thiếu sót. Em rất mong được nhận ý kiến đóng góp của các Thầy Cô để em có thể hoàn thiện hơn.

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Em xin cam đoan đây là sản phẩm đồ án của riêng tôi nhóm em và được sự hướng dẫn của thầy Lê Đình Thận. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào em xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

***MỤC LỤC***

LỜI CẢM ƠN 1

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN 3

NỘI DUNG :

1. NETWORK FLOW SỬ DỤNG THUẬT TOÁN FORD-FULKERSON
2. MINIMUM SPANNING TREE SỬ DỤNG THUẬT TOÁN PRIM
3. SINGLE SOURCE SHORTEST PATH SỬ DỤNG THUẬT TOÁN DIJKSTRA
4. **NETWORK FLOW**

ĐỊNH NGHĨA :

* Mạng ( Network ) là một [đồ thị có hướng](https://vi.wikipedia.org/wiki/%C4%90%E1%BB%93_th%E1%BB%8B_(l%C3%BD_thuy%E1%BA%BFt_%C4%91%E1%BB%93_th%E1%BB%8B)" \l "%C4%91%E1%BB%93_th%E1%BB%8B_c%C3%B3_h%C6%B0%E1%BB%9Bng" \o "Đồ thị (lý thuyết đồ thị)) G = (V, E) trong đó:

Có duy nhất một đỉnh s không có cung đi vào, được gọi là đỉnh phát (source)

Có duy nhất một đỉnh t không có cung đi ra, được gọi là đỉnh thu (sink)

Mỗi cạnh e = (u, v) ∈ E được gán một số nguyên không âm c(e) = c[u, v] và gọi là khả năng thông qua của cung đó (capacity).

Nếu mạng không có cung (u, v) thì ta thêm vào cung (u, v) với khả năng thông qua c[u,v] bằng 0.

* Luồng trên mạng : thoả mãn 2 điều kiện sau

1. **ĐK 1 (Capacity Constraint)**: Luồng trên mỗi cung e ∈ E không vượt quá khả năng thông qua của nó: **0 ≤ f(e) ≤ c(e)**
2. **ĐK 2 (Flow Conversion)**: Điều kiện cân bằng luồng trên mỗi đỉnh của mạng: Tổng luồng trên các cung vào đỉnh v bằng tổng luồng trên các cung đi ra khỏi đỉnh v

* Lát cắt nhỏ nhất :

Một lát cắt là *nhỏ nhất* nếu kích thước của nó không lớn hơn kích thước bất kì lát cắt nào khác. Ví dụ bên phải là một lát cắt nhỏ nhất: kích thước của nó là 2, và không có lát cắt nào có kích thước 1 vì đồ thị không có [cầu](https://vi.wikipedia.org/w/index.php?title=C%E1%BA%A7u_(l%C3%BD_thuy%E1%BA%BFt_%C4%91%E1%BB%93_th%E1%BB%8B)&action=edit&redlink=1" \o "Cầu (lý thuyết đồ thị) (trang chưa được viết)).

[Định lý luồng cực đại lát cắt cực tiểu](https://vi.wikipedia.org/wiki/%C4%90%E1%BB%8Bnh_l%C3%BD_lu%E1%BB%93ng_c%E1%BB%B1c_%C4%91%E1%BA%A1i_l%C3%A1t_c%E1%BA%AFt_c%E1%BB%B1c_ti%E1%BB%83u) chứng minh rằng luồng cực đại của một [đồ thị luồng](https://vi.wikipedia.org/w/index.php?title=%C4%90%E1%BB%93_th%E1%BB%8B_lu%E1%BB%93ng&action=edit&redlink=1" \o "Đồ thị luồng (trang chưa được viết)) đúng bằng tổng khả năng thông qua của lát cắt nhỏ nhất chia cắt đỉnh phát và đỉnh thu

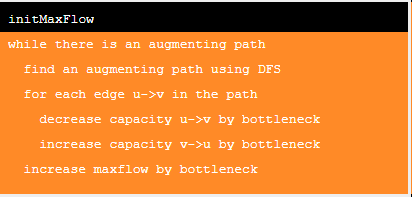
ỨNG DỤNG :

Trong thực tế , ta có thể áp dụng LUỒNG CỰC ĐẠI và LÁT CẮT CỰC TIỂU trong việc xây dựng hệ thống mạng . LUỒNG CỰC ĐẠI sẽ tối ưu hoá bandwidth của hệ thống mạng đó và LÁT CẮT CỰC TIỂU sẽ tối ưu hoá tốc độ truyền tải trong hệ thống mạng .

**THUẬT TOÁN FORD-FULKERSON**

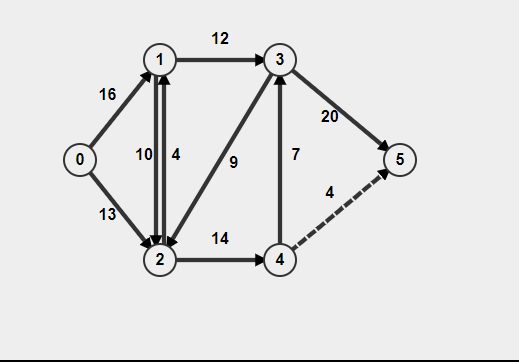
Miễn là tồn tại một đường đi từ nguồn (nút bắt đầu) đến điểm xả (nút cuối), với điều kiện tất cả các cung trên đường đi đó vẫn còn khả năng thông qua, thì ta sẽ gửi đi một luồng dọc theo đường đi đó. Sau đó chúng ta tìm một đường đi khác, và tiếp tục như vậy. Một đường đi còn khả năng thông qua là *một đường đi có khả năng mở rộng thêm* hay *một đường đi mà luồng qua đó còn khả năng tăng thêm*

* + - Input : Đồ thị *G* với khả năng thông qua *c*, một nút nguồn *s*, và một nút thoát *t*
    - Output : Luồng f sao cho f cực đại khi đi từ s tới t .

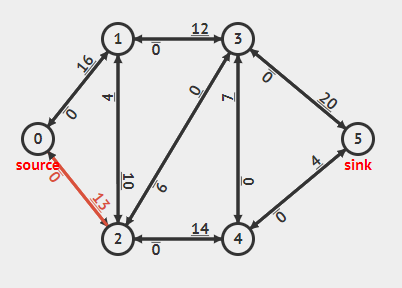


**VÍ DỤ TRÊN 1 ĐỒ THỊ CỤ THỂ**

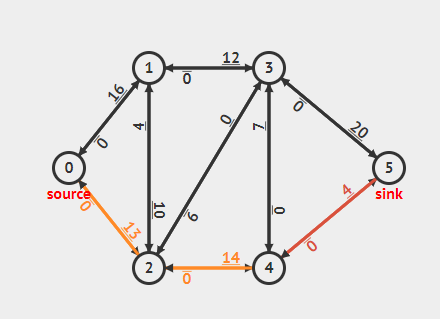
Cho đồ thị như sau :

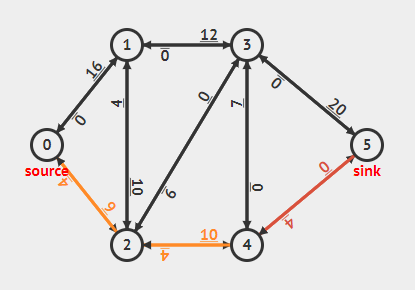


* Ta sẽ tìm Lát cắt nhỏ nhất ( Min Cut ) và Luồng cực đại ( Max Flow ) cho đồ thị trên với đỉnh nguồn là 0 và đỉnh thu là 5
* Đầu tiên khởi tạo giá trị luồng cho tất cả các cạnh là 0 , xác định 2 đỉnh nguồn và thu .

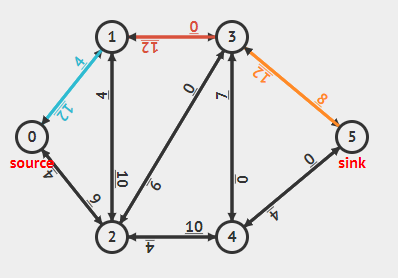


* Sau đó tìm đường từ đỉnh nguồn cho tới đỉnh thu mà tất cả các cạnh trên đó vẫn còn khả năng thông qua , đường đầu tiên 0 – 2 – 4 – 5 và ta thấy luồng có thể thông qua tối đa là 4 . Sau đó ta cập nhật khả năng thông qua trên đoạn đường từ 0 -> 2 -> 4 -> 5 += 4 , cập nhật khả năng thông qua trên đoạn đường từ 5 -> 4 -> 2 -> 0 là -= 4 . Ta có maxFlow đầu tiên = 4 .

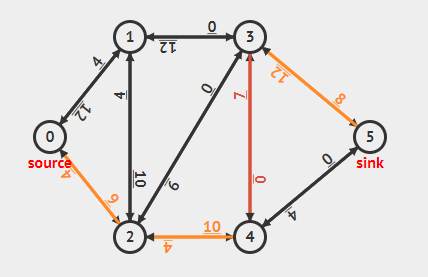




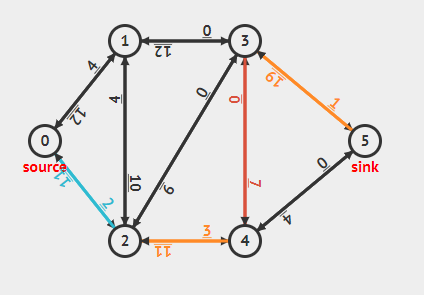
* Tương tự như cách làm trên , ta có đường đi thứ 2 từ 0 -> 1 -> 3 -> 5
* Khả năng thông qua tối đa là 12 , theo cách trên ta cập nhật luồng thông qua trên đường đi từ 0->5 và từ 5->0 sau đó có maxFlow thứ 2 là 12 .



* Trên đồ thị sau khi cập nhật vẫn còn đường đi còn khả năng thông qua nên có đường đi tiếp theo



* Khả năng thông qua = 7 , maxFlow = 7
* Cập nhật lại trên đồ thị như trên .



* Do không còn tồn tại đường đi từ 0->5 có khả năng thông qua nên thuật toán dừng lại và ta có luồng cực đại sẽ bằng maxFlow của 3 đường đi cộng lại = 4 + 12 + 7 = 23 .
* Trên hình ta có thể thấy cạnh 1-3 , cạnh 3-4 , cạnh 4-5 không còn khả năng thông qua ( khả năng thông qua = 0 ) . Đây cũng là lát cắt nhỏ nhất của đồ thị trên .
* Min Cut = {1-3 , 3-4 , 4-5 }

1. **MINIMUM SPANNING TREE**

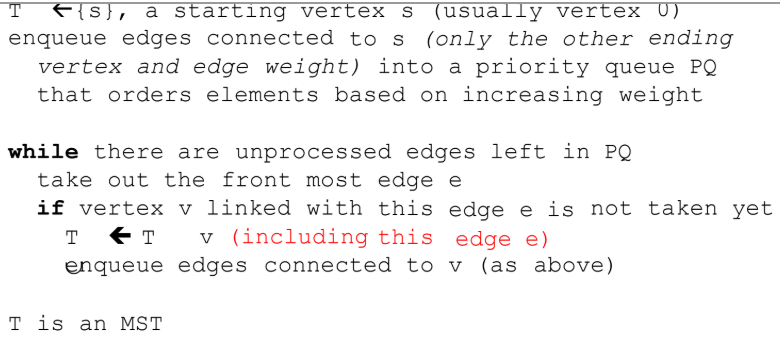
ĐỊNH NGHĨA : Cây khung nhỏ nhất của 1 đồ thị là tập hợp các cạnh trên đồ thị không tồn tại chu trình , không có đỉnh cô lập và có trọng số nhỏ nhất .

ỨNG DỤNG:

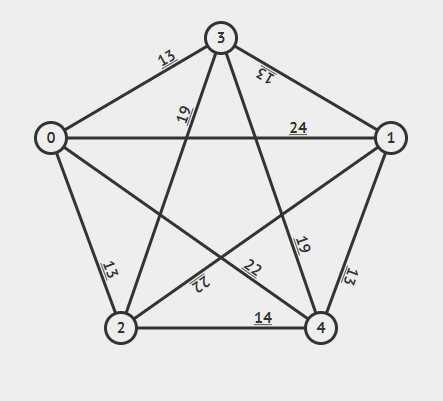
* Bài toán chọn một hành trình tiết kiệm nhất (theo tiêu chuẩn hoặc khoảng cách hoặc thời gian hoặc chi phí) trên một mạng giao thông
* Bài toán chọn một phương pháp tiết kiệm nhất để đưa ra một hệ thống động lực từ trạng thái xuất phát đến trạng một trạng thái đích
* Bài toán lập lịch thi công các công các công đoạn trong một công trình thi công lớn
* Bài toán lựa chọn đường truyền tin với chi phí nhỏ nhất trong mạng thông tin

**THUẬT TOÁN PRIM:**

* + - Input : G(V,E) , source s
    - Output : MST from s



**VÍ DỤ TRÊN 1 ĐỒ THỊ CỤ THỂ**

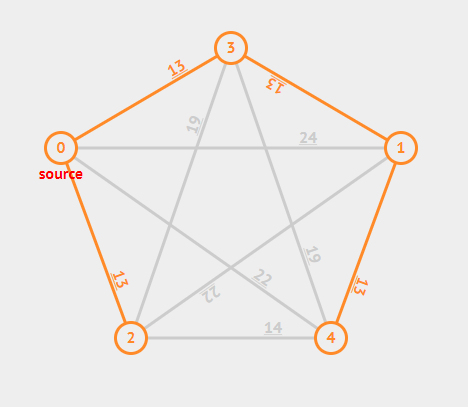


* s = 0
* Liệt kê tất cả các cạnh trong đồ thị trên và sắp xếp theo thứ tự
* Sau đó chọn tập hợp các cạnh có trọng số nhỏ nhất ( xét tất cả những cạnh từ đỉnh đang xét tới các đỉnh kề với nó ) và không tạo ra chu trình .

|  |  |  |
| --- | --- | --- |
| u | w | v |
| 0 | 24 | 1 |
| 0 | 13 | 2 |
| 0 | 13 | 3 |
| 0 | 22 | 4 |
| 1 | 22 | 2 |
| 1 | 13 | 4 |
| 2 | 19 | 3 |
| 2 | 14 | 4 |
| 3 | 13 | 1 |
| 3 | 19 | 4 |

Theo quy tắc trên , ta chọn :

* 0-> 2 : 13
* 0-> 3 : 13
* 3-> 1 : 13
* 1 -> 4 : 13
* Tất cả các đỉnh đều đã được đi qua . thuật toán dừng lại . Ta có kết quả sau :



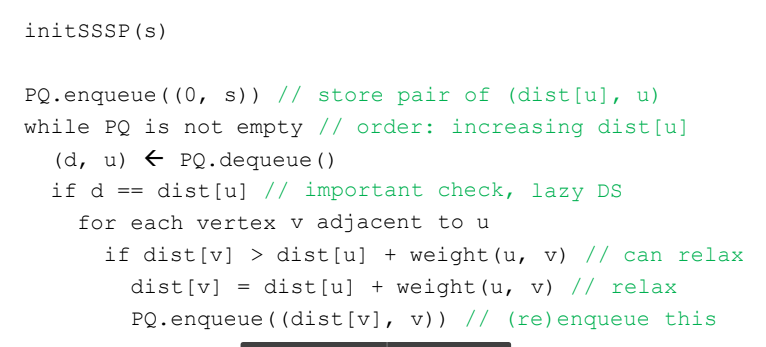
1. **SINGLE SOURCE SHORTEST PATH**

**ĐỊNH NGHĨA** : Đường đi ngắn nhất từ một nguồn đơn là đường đi giữa 2 đỉnh sao cho tổng các trọng số của các cạnh tạo nên đường đi đó là nhỏ nhất .

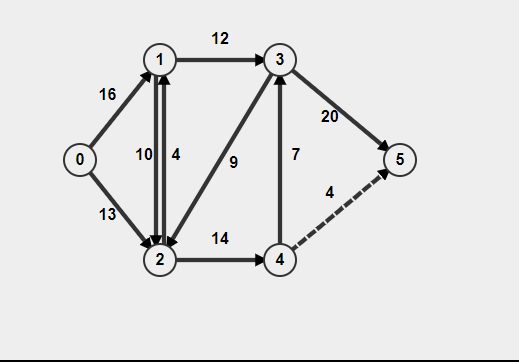
**ỨNG DỤNG** : Google Map , Xây dựng đường đi trong hệ thống mạng ….

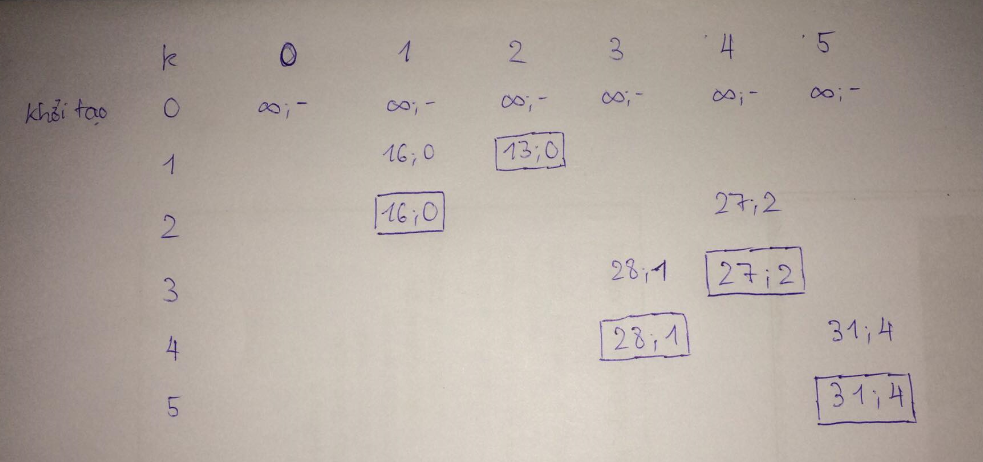
**THUẬT TOÁN DIJKSTRA**

* Input : G(V,E) , source s , terminal t
* Output : ShortestPath from s to t

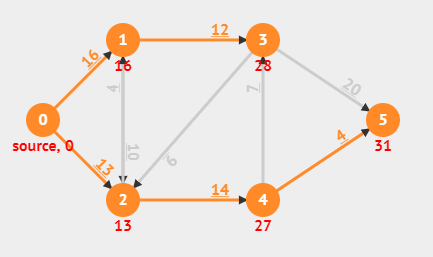


**VÍ DỤ TRÊN 1 ĐỒ THỊ CỤ THỂ**





**Kết quả :**



**HẾT**