|  |
| --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TAO TRƯỜNG ĐAI HOC SƯ PHAM TP. HỒ CHÍ MINH     HUỲNH VĂN THỌ - KHMT-17-009**  **NGUYỄN MINH NHỰT – KHMT-17-007     TÊN ĐỀ TÀI TIỂU LUẬN**  **XÁC ĐỊNH CỘNG ĐỒNG DỰA TRÊN CẤU TRÚC ĐỒ THỊ WIKIPEDIA   CHUYÊN NGÀNH KHOA HỌC MÁY TÍNH       NGƯỜI HƯỚNG DẪN: TS. BÙI THANH HÙNG** *Tp. Hồ Chí Minh, ngày 14 tháng 08 năm 2018* |

**Mục Lục**

[**1.** **Tổng quát đề tài** 1](#_Toc522038059)

[1.1. Giới thiệu chung 1](#_Toc522038060)

[1.2. Cộng đồng và cấu trúc đồ thị Wikipedia 1](#_Toc522038061)

[1.2.1 Cộng đồng WiKi 1](#_Toc522038062)

[1.2.2 Cấu trúc đồ thị 4](#_Toc522038063)

[**2.** **Giải quyết vấn đề** 6](#_Toc522038065)

[2.1. Thuật toán Prim 6](#_Toc522038066)

[2.2. Thuật toán Floyd 14](#_Toc522038067)

[2.3. Bài toán áp dụng 18](#_Toc522038068)

[2.3.1 Nguồn dữ liệu 18](#_Toc522038069)

[2.3.2 Áp dụng thuật toán để giải quyết 19](#_Toc522038070)

[**3. Kết luận** 21](#_Toc522038071)

[**4. Tài liệu tham khảo** 22](#_Toc522038072)

# **1. Tổng quát đề tài**

## 1.1. Giới thiệu chung

Để tạo nên một cộng đồng mạng không chỉ phụ thuộc vào kiến thức và khả năng của từng thành viên, mà còn phải lưu tâm về cách họ cộng tác. Trong đề tài này, chúng tôi quan tâm các hình thức tổ chức cộng đồng trong Wikipedia (gọi tắc là Wiki), cách mà cộng đồng hợp tác, liên kết, tương tác với nhau thông qua cách vận dụng cấu trúc đồ thị để kết nối người dùng thành cộng đồng trong Wikipedia. Cấu trúc đồ thị của Wikipedia bao gồm các đỉnh là các user (người dùng) và các cạnh thể hiện sự liên kết giữa người dùng với nhau thông qua việc người dùng chỉnh sửa, phản hồi, đóng góp cho trang Wikipedia của người dùng khác.

Do đó, chúng tôi quan tâm đến vấn đề rằng, thông qua việc phân tích đồ thị cấu trúc của Wikipedia liệu rằng có thể xác định được một số đặc điểm tiêu biểu của cộng đồng trong Wikipedia hay không? Cụ thể là, xác định người dùng này có tương tác với người dùng kia trực tiếp thông qua việc đóng góp trang Wiki, hay phải dựa vào việc đóng góp, thảo luận… bài viết của các thành viên khác trong cộng đồng?

Dựa vào mục tiêu trên, chúng tôi xây dựng ý tưởng xác định các đường đi giữa các đỉnh (người dùng) trong cộng đồng Wikipedia bằng thuật toán Floyd-Warshall và Prim.

Trong bài này, chung tôi tìm hiểu và trình bày theo cấu trúc như sau:

+ Giới thiệu chung về cộng đồng và cấu trúc đồ thị Wikipedia

+ Trình bày, cài đặt thuật toán Floyd Warshall và thuật toán Prim trong việc tìm đường đi ngắn nhất và tìm khung cây nhỏ nhất cho đồ thị biểu diễn bằng danh sách kề.

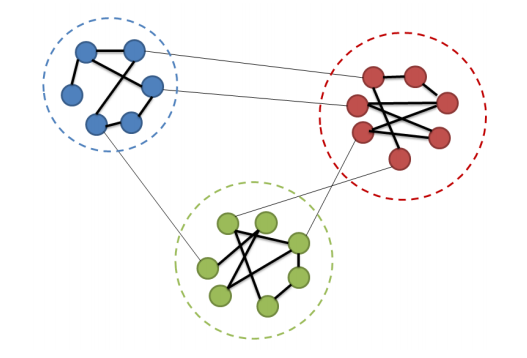
+ Vận dụng thuật toán vừa tìm hiểu vào cấu trúc đồ thị Wikipedia để xác định một số đặc điểm đã được nêu ở trên trong cộng đồng Wiki.

## 1.2. Cộng đồng và cấu trúc đồ thị Wikipedia

### **1.2.1 Cộng đồng WiKi**

Theo nghĩa thông thường, một cộng đồng là một nhóm xã hội của các cơ thể sống chung trong cùng một môi trường thường là có cùng các mối quan tâm chung. Trong cộng đồng người đó là kế hoạch, niềm tin, các mối ưu tiên, nhu cầu, nguy cơ và một số điều kiện khác có thể có và cùng ảnh hưởng đến đặc trưng và sự thống nhất của các thành viên trong cộng đồng.

Ở cấp độ thiết yếu, cộng đồng xảy ra bất cứ lúc nào hai hoặc nhiều người trở nên hài hòa (không nhất thiết phải thỏa thuận) xung quanh bất kỳ chủ đề nào. Cộng đồng phát sinh như một chức năng quan tâm và tham gia. Tại một số điểm vượt quá hai người tham gia, một cộng đồng thực sự là một cộng đồng các cộng đồng.



*Hình mô tả cộng đồng mạng*

Theo nghĩa rộng hơn, cộng đồng Wikipedia bao gồm tất cả các biên tập viên vô danh, những người ủng hộ ý thức hệ, người đọc hiện tại và thậm chí là độc giả tiềm năng của tất cả các phiên bản ngôn ngữ của Wikipedia - bách khoa toàn thư. Điều này bao gồm phần lớn dân số của trái đất, nhưng có lẽ rất quan trọng cần phải nhớ từng lần một, rất nhiều người có cổ phần trong nội dung của bách khoa toàn thư và hướng của dự án.

Một định nghĩa hẹp hơn về cộng đồng Wikipedia - vì thiếu một từ tốt hơn, hãy gọi nó là cộng đồng cộng tác viên Wikipedia - là nhóm cộng tác viên tạo ra danh tính (hoặc tài khoản người dùng hoặc IP ẩn danh thường được sử dụng) và người giao tiếp với những người đóng góp khác.

Điều gây tranh cãi là liệu có nhiều cộng đồng Wikipedia, dựa trên các ngôn ngữ khác nhau hay chỉ một ngôn ngữ . Nhiều thành viên của các cộng đồng không phải tiếng Anh cũng tham gia vào Wikipedia tiếng Anh. Các vấn đề kỹ thuật cũng được điều phối giữa tất cả các Wikipedias.

Cộng đồng Wikipedia là:

+ Đa dạng . Có rất nhiều loại người ở đây: các nhà triết học, những người lịch sử, các nhà khoa học, nghệ sĩ, những người tôn giáo, những người theo chủ nghĩa chung, các chuyên gia. Hơn nữa có sự giao tiếp giữa tất cả những loại người đó; có những nỗ lực để hiểu nhau, bất chấp sự khác biệt về ngôn ngữ hoặc văn hóa. Ngoài ra, có nhiều cách tiếp cận và sở thích vì có người.

+ Có kiến ​​thức .

+ Cá nhân . Điều này có vẻ lạ: sau khi tất cả, mục tiêu là tạo ra các mục có mục tiêu và không thiên vị cá nhân nhất có thể. Nhưng tính mở của Wikipedia cho phép tự biểu hiện hoàn toàn trong các giới hạn đó (và thậm chí không có nó trong các trang cá nhân); Wikipedians định nghĩa chính mình trong bối cảnh của dự án thông qua sở thích và mục tiêu của họ. Điều này mang lại cả lợi ích và biến chứng - Wikipedia lợi dụng các phẩm chất cá nhân như tin cậy, thấu hiểu, trí tưởng tượng, sự riêng tư và sự đồng cảm mà các tổ chức quan liêu không thể; nhưng nó không thể làm như vậy mà không có một số nhược điểm, bao gồm cả sự nhầm lẫn, thiên vị, sai lầm, và cảm xúc bị tổn thương. Một cộng đồng lành mạnh không loại trừ được vấn đề, nhưng nó hiểu cách đối phó với chúng.

+ “Mong manh” . Sự thành công của cộng đồng phụ thuộc phần lớn vào sự hiện diện của Wikipedians tốt. Sợ những yếu tố đó đi và dự án sẽ mất nhiều sức hấp dẫn của nó. Nhưng thực tế là không ai có thể chỉnh sửa hầu hết mọi thứ làm cho Wikipedia trở nên “mạnh mẽ”?

+ “Duy nhất” . Có cộng đồng nào khác ngoài kia kết hợp các thuộc tính trên không? Wikipedia có một nhiệm vụ duy nhất, hình thành cộng đồng độc đáo: nó là một wiki (và do đó rất cởi mở), nội dung mở (và do đó miễn phí), và một dự án bách khoa toàn thư. Không có gì giống như thế này trên thế giới.

Cộng đồng đóng một vai trò quan trọng trong việc giúp Wikipedia-the-project thực hiện Wikipedia - bách khoa toàn thư. Bởi vì Wikipedia là một wiki, nên rất nhiều thông tin liên lạc và cộng tác là cần thiết để tạo ra bách khoa toàn thư. Nhiều tay làm cho công việc nhẹ, nhưng phải có sự phối hợp giữa tất cả những bàn tay đó để làm cho công việc xảy ra chút nào. Vai trò của cộng đồng, như một loại siêu thực thể khoa học viễn tưởng, là:

+ Tổ chức và chỉnh sửa các trang riêng lẻ

+ Điều hướng cấu trúc giữa các trang

+ Giải quyết xung đột giữa các thành viên riêng lẻ

+ Tự thiết kế lại - tạo ra các quy tắc và các mẫu hành vi

Những gì không phải là một cộng đồng Wikipedia

+ Wikipedia không phải là một trận chiến: Trong cộng đồng Wikipedia, mọi người đối xử với nhau một cách hòa nhã với thái độ hợp tác. Không lăng mạ, bôi nhọ hoặc dùng các lời lẽ thiếu văn hóa đối với những người không có cùng quan điểm với mình. Hơn thế nữa, các thành viên được khuyến khích giải quyết các vấn đề bằng các biện pháp có học, với những lời thảo luận nhã nhặn. Xin đừng mở mới hoặc sửa đổi một mục từ chỉ vì để chứng minh các luận điểm của mình. Cũng không được có những hành vi đe dọa dưới bất kỳ hình thức nào đối với bất kỳ thành viên nào của Wikipedia hay Wikimedia Foundation 3. Bất cứ lời đe doạ nào cũng sẽ bị xóa bỏ và thành viên đó có thể bị treo quyền sử dụng.

+ Wikipedia không phải là nơi thử nghiệm về tình trạng vô chính phủ Wikipedia được biết đến bởi yếu tố mở và tự do, tuy nhiên sự tự do và tính mở này giới hạn trong mục đích xây dựng một bộ từ điển bách khoa. Do vậy, Wikipedia không phải là một diễn đàn dành cho các bài diễn văn về tự do ngoài luật định. Thực tế, Wikipedia là một dự án mở và tự quản lý, điều này không có nghĩa dùng để phát triển các cộng đồng vô chính phủ. Mục đích của chúng ta là xây dựng từ điển bách khoa, không phải để thử nghiệm các giới hạn của thuyết vô chính phủ. Nếu bạn muốn thử nghiệm điều này thì xin hãy đến Anarchopedia.

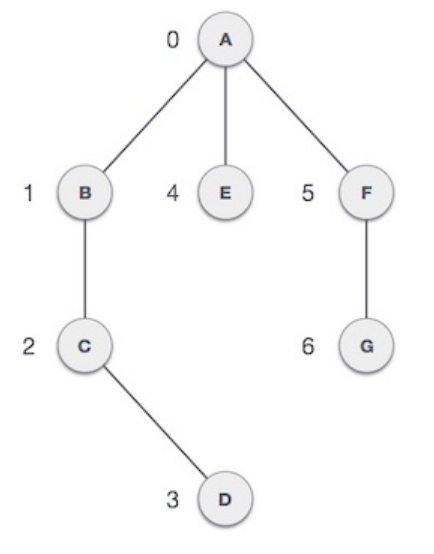
+ Wikipedia không phải là nơi thử nghiệm về dân chủ: Wikipedia cũng không phải là nơi thử nghiệm mô hình dân chủ. Phương thức chủ yếu và được khuyến khích là tìm kiếm sự đồng thuận thông qua thảo luận và giải quyết những mâu thuẫn, chứ không phải là biểu quyết. Do vậy, việc lấy các ý kiến của đa số không thể coi là nguyên tắc hoạt động của Wikipedia. Tuy nhiên, các biểu quyết theo mô hình dân chủ trực tiếp (với đa số phiếu thắng thiểu số) trong Wikipedia vẫn thường xuyên được sử dụng, kết quả của những biểu quyết này thường chỉ là một trong những giải pháp để đưa ra quyết định cuối cùng. Những thảo luận bên trong từng biểu quyết là cực kỳ quan trọng để thu nhận được sự đồng thuận. Ví dụ, một hoạt động rất quan trọng trong Wikipedia là đánh giá xem bài viết nào không thuộc Wikipedia thông qua mục Wikipedia:Biểu quyết xoá bài. Trang này cần phải được hiểu là những thảo luận xung quanh việc giữ hay xoá còn quan trọng hơn nhiều so với số phiếu cao nhất.

+ Wikipedia không quan liêu: Trong quá trình Wikipedia hoạt động, khá nhiều các nguyên tắc, quy định và bán quy định được cộng đồng người sử dụng Wikipedia xây dựng để định hướng chung, hạn chế các thảo luận không cần thiết. Tuy nhiên, các quy định đấy không phải là bất biến, chúng có thể được thảo luận và viết lại bất kỳ lúc nào. Trong quá trình thảo luận để giải quyết các bất đồng, việc dẫn chứng các nguyên tắc, quy định của Wikipedia chỉ là một trong nhiều cách lập luận, phân tích. Do đó, khi xây dựng những nguyên tắc và quy định cũng cần tránh hoặc giảm thiểu sự cứng nhắc.....

### **1.2.2 Cấu trúc đồ thị**

Một đồ thị (đồ thị) là một dạng biểu diễn hình ảnh của một tập các đối tượng, trong đó các cặp đối tượng được kết nối bởi các link. Các đối tượng được nối liền nhau được biểu diễn bởi các điểm được gọi là **các đỉnh (vertices)**, và các link mà kết nối các đỉnh với nhau được gọi là **các cạnh (edges)**.

Các hình toán học có thể được biểu diễn trong cấu trúc dữ liệu. Chúng ta có thể biểu diễn một hình bởi sử dụng một mảng các đỉnh và một mảng hai chiều của các cạnh. Trước khi tiếp tục, chúng ta tìm hiểu một vài khái niệm quan trọng sau:

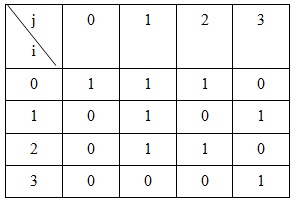
* **Đỉnh (Vertex)**: Mỗi nút của hình được biểu diễn như là một đỉnh. Trong ví dụ dưới đây, các hình tròn biểu diễn các đỉnh. Do đó, các điểm từ A tới G là các đỉnh. Chúng ta có thể biểu diễn các đỉnh này bởi sử dụng một mảng, trong đó đỉnh A có thể được nhận diện bởi chỉ mục 0, điểm B là chỉ mục 1, … như hình dưới.
* **Cạnh (Edge)**: Cạnh biểu diễn một đường nối hai đỉnh. Trong hình dưới, các đường nối A và B, B và C, … là các cạnh. Chúng ta có thể sử dụng một mảng hai chiều để biểu diễn các cạnh này. Trong ví dụ dưới, AB có thể được biểu diễn như là 1 tại hàng 0; BC là 1 tại hàng 1, cột 2, …
* **Kề nhau**: Hai đỉnh là kề nhau nếu chúng được kết nối với nhau thông qua một cạnh. Trong hình dưới, B là kề với A; C là kề với B, …
* **Đường**: Đường biểu diễn một dãy các cạnh giữa hai đỉnh. Trong hình dưới, ABCD biểu diễn một đường từ A tới D.

Các bài toán đồ thị: Tìm đồ thị con, tô màu đồ thị, bài toán đường đi, [luồng](https://vi.wikipedia.org/wiki/Lu%E1%BB%93ng_tr%C3%AAn_m%E1%BA%A1ng)

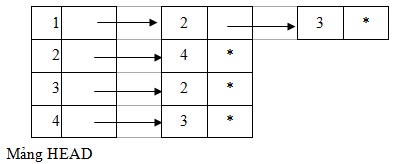
Một số cấu trúc dữ liệu có thể dùng để biểu diễn đồ thị. Việc chọn cấu trúc dữ liệu nào là tuỳ thuộc vào các phép toán trên các cung và đỉnh của đồ thị. Hai cấu trúc thường gặp là biểu diễn đồ thị bằng ma trận kề (adjacency matrix) và biểu diễn đồ thị bằng danh sách các đỉnh kề (adjacency list).

**Biểu diễn đồ thị bằng ma trận kề**

Ta cũng có thể biểu diễn true là 1 còn false là 0. Với cách biểu diễn này thì đồ thị có ma trận kề như sau:

****

**Biểu diễn đồ thị bằng danh sách các đỉnh kề:**

Trong cách biểu diễn này, ta sẽ lưu trữ các đỉnh kề với một đỉnh i trong một danh sách liên kết theo một thứ tự nào đó để biểu diễn cho đồ thị có n đỉnh. HEAD[i] là con trỏ trỏ tới danh sách các đỉnh kề với đỉnh i được biểu diễn như sau:

**Cấu trúc đồ thị được vận dụng trong wikipedia:**

* Cấu trúc đồ thị được sử dụng trong wikipedia là một cấu trúc mang tính xã hội được cấu tạo từ các nút và các cung trong đó các nút được liên kết với nhau bởi 1 hoặc nhiều cung.
* Mỗi nút (cá nhân) biểu diễn cho 1 đối tượng: một người dùng, một tài liệu, một thông tin, hình ảnh, văn bản, bài báo,…
* Liên kết giữa các nút có thể là quan hệ sử dụng, đọc, tham khảo, có mối liên hệ, được chỉnh sửa, chú thích…
* Cấu trúc đồ thị được sử dụng trong wikipedia được biểu diễn bởi 2 cấu trúc phổ biến : danh sách đỉnh kề và ma trận kề. Để tính toán và phân tích các liên kết trong mạng, thường ưu tiên áp dụng lý thuyết đồ thị do đồ thị là 1 trong những công cụ hữu hiệu để thể hiện các thông tin, tài liệu, người dùng,…Để tìm kiếm các thông tin chính xác hoặc gần nhất một cách nhanh chóng, tránh việc lãng phí thời gian với những thông tin, tài liệu không cần thiết.
  + Các đỉnh được dùng để biểu diễn các nút
  + Các cạnh dùng để biểu diễn liên kết giữa các nút, các cạnh có thể có hướng hoặc vô hướng và có thể được đánh trọng số.

# **2. Giải quyết vấn đề**

## 2.1. Thuật toán Prim

**Thuật toán Prim** là một [thuật toán tham lam](https://vi.wikipedia.org/wiki/Gi%E1%BA%A3i_thu%E1%BA%ADt_tham_lam) để tìm [cây bao trùm nhỏ nhất](https://vi.wikipedia.org/wiki/C%C3%A2y_bao_tr%C3%B9m_nh%E1%BB%8F_nh%E1%BA%A5t) của một [đồ thị vô hướng](https://vi.wikipedia.org/wiki/%C4%90%E1%BB%93_th%E1%BB%8B_v%C3%B4_h%C6%B0%E1%BB%9Bng) [có trọng số](https://vi.wikipedia.org/w/index.php?title=%C4%90%E1%BB%93_th%E1%BB%8B_c%C3%B3_tr%E1%BB%8Dng_s%E1%BB%91&action=edit&redlink=1) [liên thông](https://vi.wikipedia.org/wiki/%C4%90%E1%BB%93_th%E1%BB%8B_li%C3%AAn_th%C3%B4ng). Nghĩa là nó tìm một tập hợp các [cạnh](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)) của đồ thị tạo thành một [cây](https://vi.wikipedia.org/wiki/C%C3%A2y_(l%C3%BD_thuy%E1%BA%BFt_%C4%91%E1%BB%93_th%E1%BB%8B)) chứa tất cả các [đỉnh](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)), sao cho tổng trọng số các cạnh của cây là nhỏ nhất.

Thuật toán được tìm ra năm 1930 bởi [nhà toán học](https://vi.wikipedia.org/wiki/Danh_s%C3%A1ch_nh%C3%A0_to%C3%A1n_h%E1%BB%8Dc) [người Séc](https://vi.wikipedia.org/wiki/Ng%C6%B0%E1%BB%9Di_S%C3%A9c) [Vojtěch Jarník](https://vi.wikipedia.org/w/index.php?title=Vojt%C4%9Bch_Jarn%C3%ADk&action=edit&redlink=1) và sau đó bởi nhà nghiên cứu khoa học máy tính [Robert C. Prim](https://vi.wikipedia.org/w/index.php?title=Robert_C._Prim&action=edit&redlink=1) năm 1957 và một lần nữa độc lập bởi [Edsger Dijkstra](https://vi.wikipedia.org/wiki/Edsger_Dijkstra) năm 1959.

***Mô tả thuật toán***:

Thuật toán xuất phát từ một cây chỉ chứa đúng một đỉnh và mở rộng từng bước một, mỗi bước thêm một cạnh mới vào cây, cho tới khi bao trùm được tất cả các đỉnh của đồ thị.

* Dữ liệu vào: Một đồ thị có trọng số liên thông với tập hợp đỉnh *V* và tập hợp cạnh *E* (trọng số có thể âm). Đồng thời cũng dùng *V* và *E* để ký hiệu số đỉnh và số cạnh của đồ thị.
* Khởi tạo: *V*mới = {*x*}, trong đó *x* là một đỉnh bất kì (đỉnh bắt đầu) trong *V*, *E*mới = {}
* Lặp lại cho tới khi *V*mới = *V*:
  + Chọn cạnh (*u*, *v*) có trọng số nhỏ nhất thỏa mãn *u* thuộc *V*mới và *v* không thuộc *V*mới (nếu có nhiều cạnh như vậy thì chọn một cạnh bất kì trong chúng)
  + Thêm *v* vào *V*mới, và thêm cạnh (*u*, *v*) vào *E*mới
* Dữ liệu ra: *V*mới và *E*mới là tập hợp đỉnh và tập hợp cạnh của một cây bao trùm nhỏ nhất

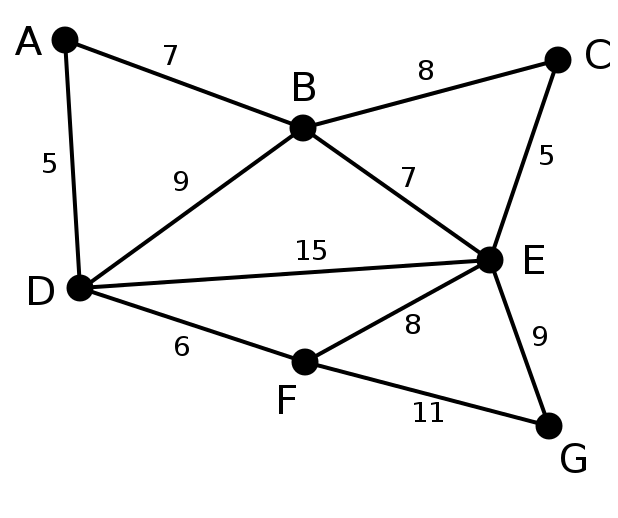
**Thuật toán Prim**  
Cho G = (X, E) là một đồ thị liên thông có trọng số gồm n đỉnh. Thuật toán Prim  
được dùng để tìm ra cây khung nhỏ nhất của G.

• Bước 1: Chọn tùy ý x0 thuộc X và khởi tạo V:= { x0 }; T := Ø. Trong đó  
X là tập các đỉnh của đồ thị, V là tập các đỉnh được chọn vào cây khung  
nhỏ nhất và T là tập các cạnh của cây này.

• Bước 2: Trong số những cạnh nối đỉnh x với đỉnh y mà x ∈ V và y ∈X\V ta chọn cạnh e có trọng số nhỏ nhất. Nếu không có cạnh e thỏa yêu  
câu: DỪNG (1)

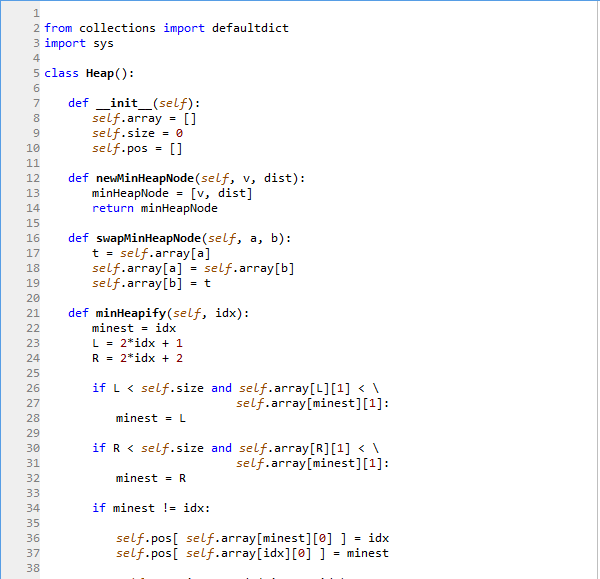
• Bước 3: Thêm đỉnh y vào tập V và thêm cạnh e vào tập T.

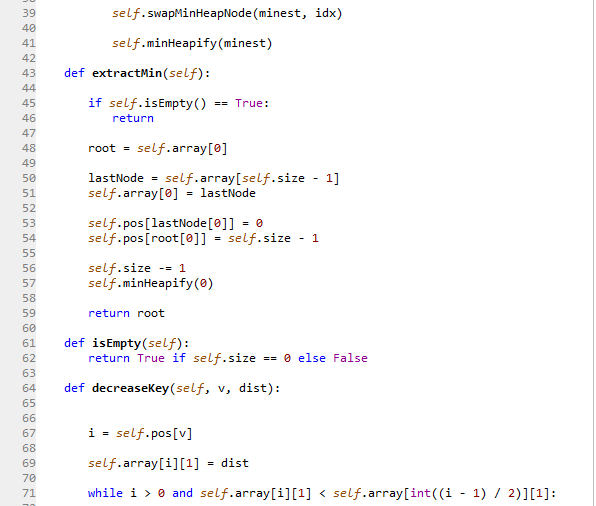
• Bước 4: Nếu T đủ n – 1 phần tử thì DỪNG (2), ngược lại làm tiếp tục  
bước 2.

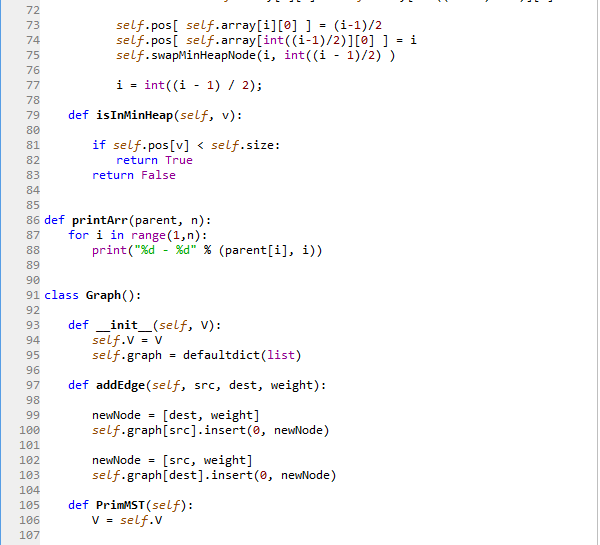
**Ví dụ minh họa**: Tìm khung cây nhỏ nhất của bài toán sau:

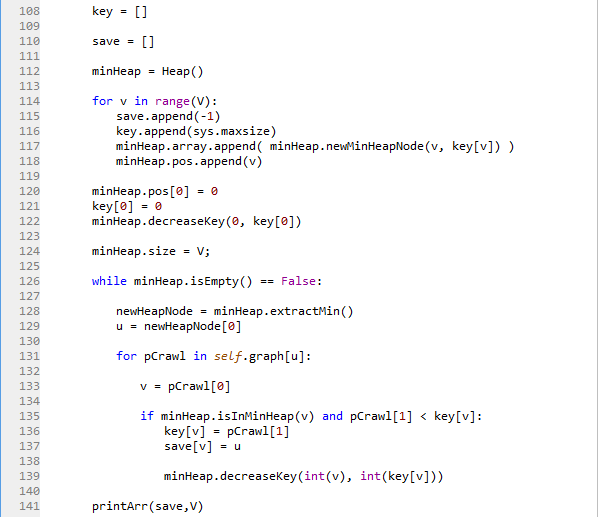
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hình minh họa** | **U** | **Cạnh (u,v)** | **V \ U** | **Mô tả** |
| [Prim Algorithm 0.svg](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_tin:Prim_Algorithm_0.svg) | {} |  | {A,B,C,D,E,F,G} | Đây là đồ thị có trọng số ban đầu. Các số là các trọng số của các cạnh. |
| [Prim Algorithm 1.svg](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_tin:Prim_Algorithm_1.svg) | {D} | (D,A) = 5 **V**  (D,B) = 9 (D,E) = 15 (D,F) = 6 | {A,B,C,E,F,G} | Chọn một cách tùy ý đỉnh **D** là đỉnh bắt đầu. Các đỉnh **A**, **B**, **E** và **F** đều được nối trực tiếp tới **D** bằng cạnh của đồ thị. **A** là đỉnh gần **D** nhất nên ta chọn **A** là đỉnh thứ hai của cây và thêm cạnh **AD** vào cây. |
| [Prim Algorithm 2.svg](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_tin:Prim_Algorithm_2.svg) | {A,D} | (D,B) = 9 (D,E) = 15 (D,F) = 6 **V** (A,B) = 7 | {B,C,E,F,G} | Đỉnh được chọn tiếp theo là đỉnh gần **D** hoặc **A** nhất. **B** có khoảng cách tới **D** bằng 9 và tới **A** bằng 7, **E** có khoảng cách tới cây hiện tại bằng 15, và **F** có khoảng cách bằng 6. **F** là đỉnh gần cây hiện tại nhất nên chọn đỉnh **F** và cạnh **DF**. |
| [Prim Algorithm 3.svg](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_tin:Prim_Algorithm_3.svg) | {A,D,F} | (D,B) = 9 (D,E) = 15 (A,B) = 7 **V** (F,E) = 8 (F,G) = 11 | {B,C,E,G} | Thuật toán tiếp tục tương tự như bước trước. Chọn đỉnh **B** có khoảng cách tới **A** bằng 7. |
| [Prim Algorithm 4.svg](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_tin:Prim_Algorithm_4.svg) | {A,B,D,F} | (B,C) = 8 (B,E) = 7 **V** (D,B) = 9 chu trình (D,E) = 15 (F,E) = 8 (F,G) = 11 | {C,E,G} | Ở bước này ta chọn giữa **C**, **E**, và **G**. **C** có khoảng cách tới **B** bằng 8, **E** có khoảng cách tới **B** bằng 7, và **G** có khoảng cách tới **F** bằng 11. **E** là đỉnh gần nhất, nên chọn đỉnh **E** và cạnh **BE**. |
| [Prim Algorithm 5.svg](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_tin:Prim_Algorithm_5.svg) | {A,B,D,E,F} | (B,C) = 8 (D,B) = 9 chu trình (D,E) = 15 chu trình (E,C) = 5 **V** (E,G) = 9 (F,E) = 8 chu trình (F,G) = 11 | {C,G} | Ở bước này ta chọn giữa **C** và **G**. **C** có khoảng cách tới **E** bằng 5, và **G** có khoảng cách tới **E** bằng 9. Chọn **C** và cạnh **EC**. |
| [Prim Algorithm 6.svg](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_tin:Prim_Algorithm_6.svg) | {A,B,C,D,E,F} | (B,C) = 8 chu trình (D,B) = 9 chu trình (D,E) = 15 chu trình (E,G) = 9 **V** (F,E) = 8 chu trình (F,G) = 11 | {G} | Đỉnh **G** là đỉnh còn lại duy nhất. Nó có khoảng cách tới **F** bằng 11, và khoảng cách tới **E** bằng 9. **E** ở gần hơn nên chọn đỉnh **G** và cạnh **EG**. |
| [Prim Algorithm 7.svg](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_tin:Prim_Algorithm_7.svg) | {A,B,C,D,E,F,G} | (B,C) = 8 chu trình (D,B) = 9 chu trình (D,E) = 15 chu trình (F,E) = 8 chu trình (F,G) = 11 chu trình | {} | Hiện giờ tất cả các đỉnh đã nằm trong cây và [cây bao trùm nhỏ nhất](https://vi.wikipedia.org/wiki/C%C3%A2y_bao_tr%C3%B9m_nh%E1%BB%8F_nh%E1%BA%A5t) được tô màu xanh lá cây. Tổng trọng số của cây là 39. |

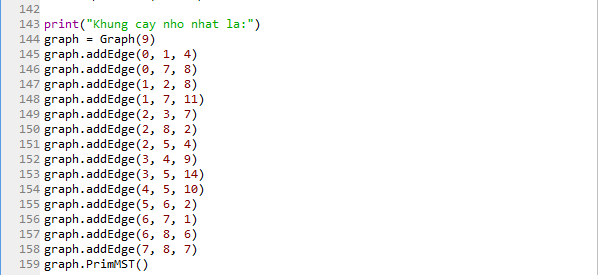
**Code python**









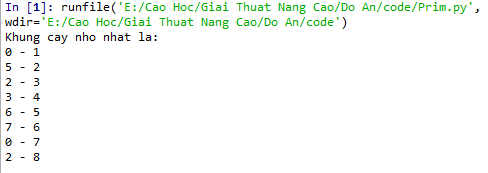


**Input**

**Số đỉnh: 9**

|  |  |  |
| --- | --- | --- |
| **User 1** | **User 2** | **Weight** |
| **0** | **1** | **4** |
| **0** | **7** | **8** |
| **1** | **2** | **8** |
| **1** | **7** | **11** |
| **2** | **3** | **7** |
| **2** | **8** | **2** |
| **2** | **5** | **4** |
| **3** | **4** | **9** |
| **3** | **5** | **14** |
| **4** | **5** | **10** |
| **5** | **6** | **2** |
| **6** | **7** | **1** |
| **6** | **8** | **6** |
| **7** | **8** | **7** |

**Output**

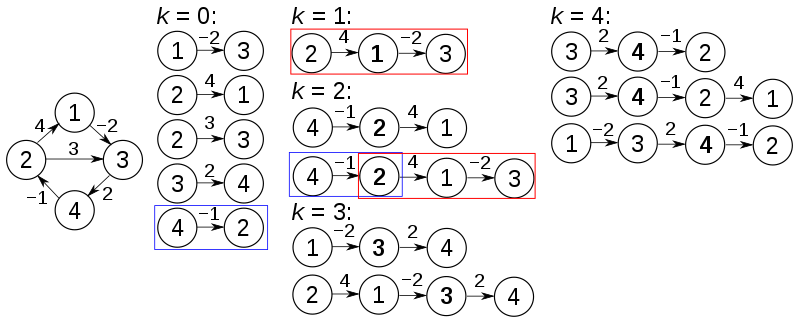


## 2.2. Thuật toán Floyd

Thuật toán Floyd Warshall được Robert Floyd công bố vào năm 1962. Tuy nhiên về cơ bản thì khi ấy, nhìn chung nó giống như thuật toán được Bernard Roy công bố năm 1959. Vào năm 1962, Stephen Warshall đã chuyển dooirdd từ automation hữu hạn thành một biểu thức chính quy. Công thức hiện tại của thuật toán với bâ vòng lặp lồng nhau được Peter Ingerman vào năm 1962. Do đó, thuật toán còn được gọi là thuật toán Floyd , các thuật toán Roy-Warshall , các thuật toán Roy-Floyd , hoặc các thuật toán WFI .

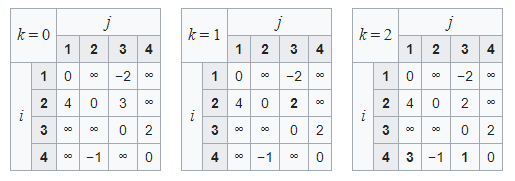
Thuật toán Floyd-Warshall sẽ tìm đường đi ngắn nhất giữa mọi đỉnh sau một lần chạy thuật toán. Một tính chất nữa là Floyd-Warshall có thể chạy trên đồ thị có các cạnh có trọng số có thể âm, tức là không bị giới hạn như Dijkstra. Tuy nhiên, lưu ý là trong đồ thị không được có vòng (cycle) nào có tổng các cạnh là âm, nếu có vòng như vậy ta không thể tìm được đường đi ngắn nhất (mỗi lần đi qua vòng này độ dài quãng đường lại giảm, nên ta có thể đi vô hạn lần)

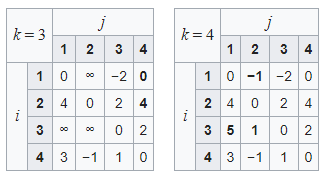
**Mô tả thuật toán**:

Ví dụ minh họa: 

Trước vòng lặp đầu tiên của vòng lặp ngoài, được gắn nhãn k = 0 ở trên, các đường dẫn duy nhất được biết tương ứng với các cạnh đơn trong biểu đồ. Tại k = 1 , các đường đi qua đỉnh 1 được tìm thấy: đặc biệt, đường dẫn [2,1,3] được tìm thấy, thay thế đường dẫn [2,3] có ít cạnh hơn nhưng dài hơn (về mặt trọng lượng ). Tại k = 2 , các đường đi qua các đỉnh {1,2} được tìm thấy. Các hộp màu đỏ và màu xanh cho thấy cách đường dẫn [4,2,1,3] được lắp ráp từ hai đường dẫn đã biết [4,2] và [2,1,3] đã gặp trong các lần lặp trước, với 2 trong giao lộ. Đường dẫn [4,2,3] không được xem xét, vì [2,1,3] là con đường ngắn nhất gặp phải từ 2 đến 3. Tại k = 3, các đường đi qua các đỉnh {1,2,3} được tìm thấy. Cuối cùng, tại k = 4 , tất cả các đường đi ngắn nhất được tìm thấy.

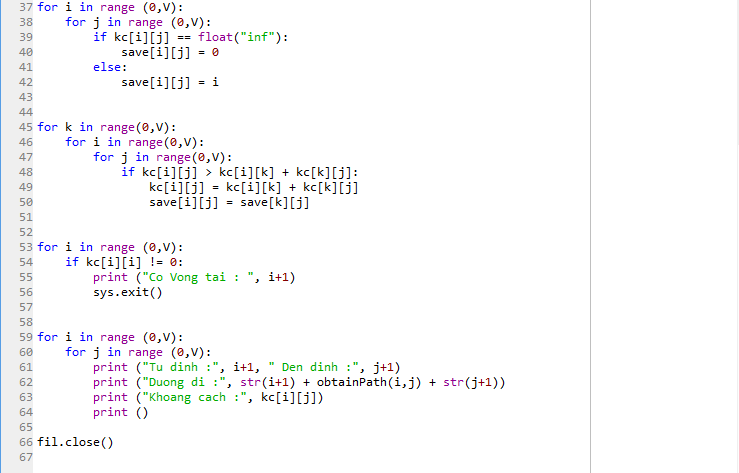
Ma trận khoảng cách tại mỗi lần lặp của k , với khoảng cách được cập nhật bằng chữ in đậm , sẽ là:



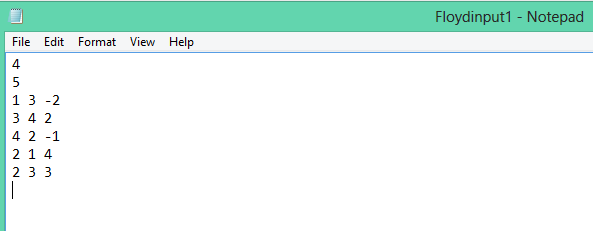


**Code python**:

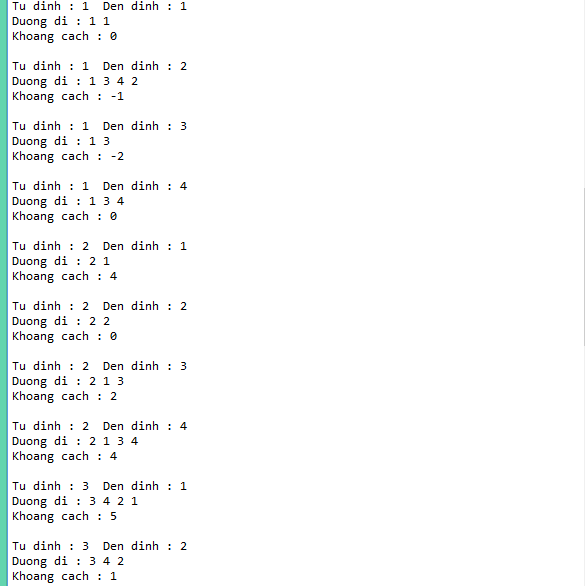


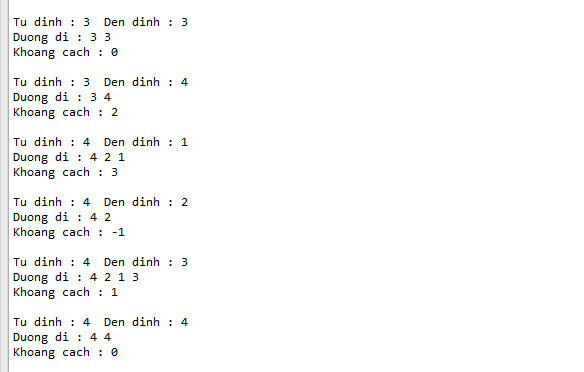


**Input**:



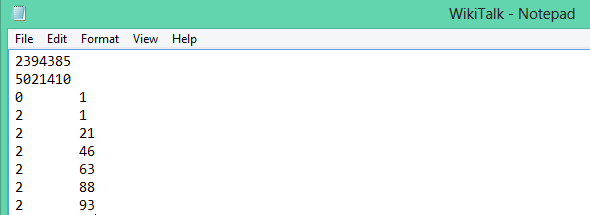
**Output**:





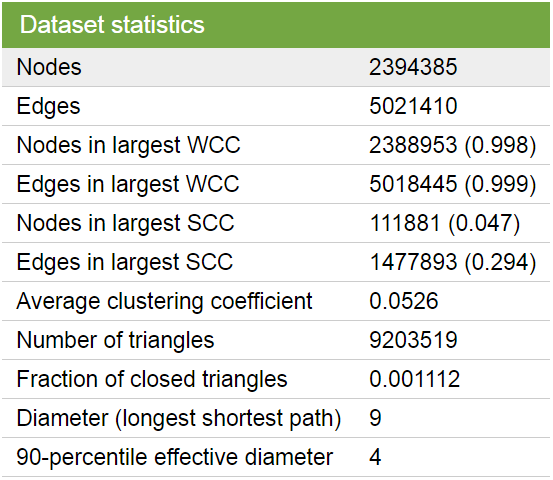
## 2.3. Bài toán áp dụng

### **2.3.1 Nguồn dữ liệu**

Wikipedia là một bách khoa toàn thư miễn phí được viết bởi các tình nguyện viên trên khắp thế giới. Mỗi người dùng đã đăng ký có một trang thảo luận, người dùng này và những người dùng khác có thể chỉnh sửa để giao tiếp và thảo luận các cập nhật cho các bài viết khác nhau trên Wikipedia. Sử dụng bản hoàn chỉnh mới nhất của lịch sử sửa đổi trang Wikipedia (từ ngày 3 tháng 1 năm 2008), trích xuất tất cả các thay đổi của trang thảo luận của người dùng và tạo một mạng. Các nút trong mạng đại diện cho người dùng Wikipedia và một cạnh từ nút i đến nút j đại diện cho người dùng đó ít nhất một lần i đã chỉnh sửa một trang thảo luận của người dùng j.

*Hình mô tả dữ liệu Wikipedia (2008)*

Theo hình trên, số đỉnh (số user – người dùng) là 2394385, số cạnh (tương tác giữa các user) là 5021410. Người dùng “0” giao tiếp với người dùng “1”, người dùng “2” giao tiếp với người dùng “1”, người dùng “2” giao tiếp với người dùng “21”…



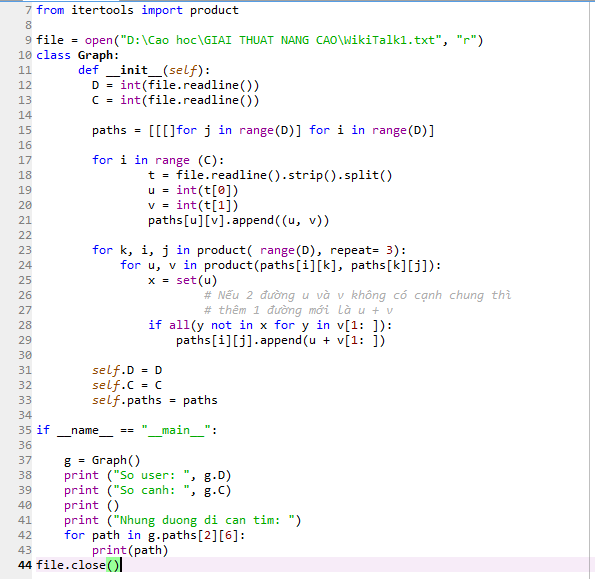
Link dữ liệu trên trang Snap của đại học Stanford: <https://snap.stanford.edu/data/wiki-Talk.html>

### **2.3.2 Áp dụng thuật toán để giải quyết**

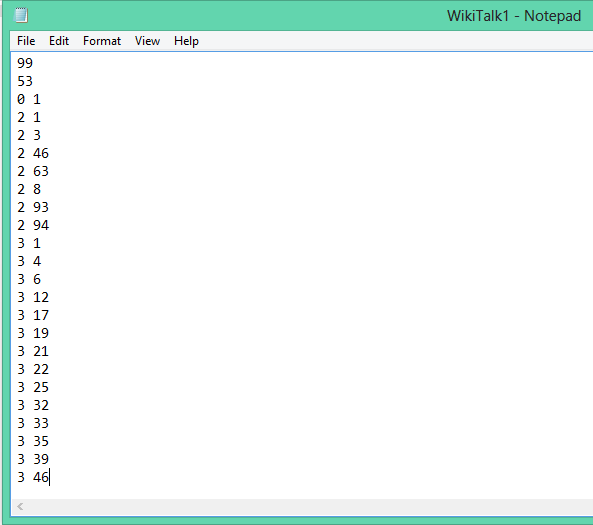
Thuật toán Floyd và Prim thường dùng được tìm đường đi ngắn nhất giữa các đỉnh và tìm cây khung nhỏ nhất trong đồ thị có trong số. Tuy nhiên, Floyd và Prim còn có thể được dùng để liệt kê tất cả các đường đi đơn (mỗi đỉnh đi qua nhiều nhất một lần) trong đồ thị không trọng số như dữ liệu Wiki-Talk ở trên. Từ đó, ta có thể dựa vào Floyd và Prim để tìm xem người dùng này có giao tiếp (chỉnh sửa, đóng góp bài viết) người dùng khác hay không. Hoặc là kiểm xem mối liên hệ giữa các user thông qua việc tương các user với nhau. Ý tưởng Floyd như sau:

* Kí hiệu (a, b, c) là một đường đi bắt đầu từ a, đi qua b rồi đến c.
* Gọi paths[u][v] là danh sách tất cả các đường đi từ đỉnh u đến đỉnh v.
* Ban đầu, nếu u và v không có cạnh nối thì paths[u][v] là danh sách rỗng, ngược lại ta thêm một đường đi trực tiếp từ u đến v vào danh sách này. Vậy, nếu u và v có cạnh nối thì paths[u][v] = [(u, v)].
* Duyệt 3 vòng for (k, i, j) như trong Floyd bình thường.
* Đến đây, ta thấy kết hợp paths[i][k] và paths[k][j] lại sẽ cho ta thêm những đường đi từ i đến j, qua trung gian là k.
* Cần chú ý kiểm tra xem 2 đường đi có đỉnh trùng không trước khi kết hợp chúng.

***Chương trình:***



***Input:***



***Output:***



# **3. Kết luận**

**Kết quả đạt được của đề tài**:

Trong đề tài này, chúng tôi đã tìm hiểu được cộng đồng Wikipedia và cấu trúc đồ thị của nó. Bên cạch đó, chúng tôi hiểu rõ thêm về hai thuật toán: Floyd-Warshall và Prim trong công việc hỗ trợ phân tích đồ thị. Thuật toán được sử dụng nhiều trong việc tìm đường đi ngắn nhất (Floyd-Warshall) và tìm câu khung tối tiểu (Prim) với đồ thị có trọng số. Tuy nhiên, trong bài toán đặt ra ở trên, hai thuật toán này vẫn có thể áp dụng được vào việc phân tích đồ thị Wikipedia (đồ thị không trọng số) và thu được kết quả mong muốn, tìm ra được mối liên hệ giữa mọi người dùng với nhau trong cộng đồng Wikipedia.

**Định hướng tương lai:**

Tuy thu được kết quả mong muốn nhưng đề tài vẫn còn hạn chế. Do Floyd-Warshall và Prim hoạt động trong thời gian O(V3), O(V2) nên nó không hiệu quả trong việc xử lí dữ liệu lớn như cộng đồng Wikipedia ( nếu có khoảng 2PB dữ liệu thì việc thực thi tốn vài năm). Nếu đồ thị biểu diễn cộng đồng là một đồ thị thưa thì việc áp dụng này rất khó khăn, gần như không tìm được mối liên kết. Từ đó, chúng tôi định hướng phát triển đề tài bằng việc tìm hiểu các thuật toán khác để hỗ trợ trong việc xử lí dữ liệu lớn.

# **4. Tài liệu tham khảo**

1. <http://www2009.eprints.org/191/1/p1221.pdf>

2. <http://snap.stanford.edu/class/cs224w-2010/proj2010/08_final_paper.pdf>

3.https://github.com/SergiuTripon/msc-thesis-na-epsrc/wiki/Community-Detection-(Wikipedia)

4.http://www.science.smith.edu/classwiki/images/a/a9/Network\_analysis\_collaboration\_wikipedia.pdf

5. <https://snap.stanford.edu/data/wiki-Talk.html>

6. <https://github.com/Robert-Campbell-256/Number-Theory-Python>

7. <https://vietcodes.github.io/algo/floyd>