

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM  
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG  
KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI BÁO CÁO CUỐI KỲ MÔN  
TOÁN TỔ HỢP VÀ ĐỒ THỊ**

# **PHÂN TÍCH MẠNG XÃ HỘI VÀ CÁC GIẢI PHÁP HIỆN CÓ**

*Giảng viên hướng dẫn:* **Thầy BÙI DANH DƯƠNG**

*Sinh viên thực hiện:* **TÔ VĨNH KHANG - 51800408**

**Lớp : 18050203**

**Khoá : 22**

**Nhóm : 10**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020**

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM  
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG  
KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI BÁO CÁO CUỐI KỲ MÔN  
TOÁN TỔ HỢP VÀ ĐỒ THỊ**

# **PHÂN TÍCH MẠNG XÃ HỘI VÀ CÁC GIẢI PHÁP HIỆN CÓ**

*Giảng viên hướng dẫn:* **Thầy BÙI DANH HƯỜNG**

*Sinh viên thực hiện:* **TÔ VĨNH KHANG - 51800408**

**Lớp : 18050203**

**Khoá : 22**

**Nhóm : 10**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020**

## LỜI CẢM ƠN

Em xin chân thành cảm ơn Khoa Công nghệ thông tin và Trường Đại học Tôn Đức Thắng đã tạo điều kiện cho chúng em được học tập trong suốt thời gian qua. Chân thành cảm ơn Thầy Bùi Danh Hường đã hướng dẫn trong quá trình học tập môn Toán tổ hợp và Đồ thị. Giúp em có thêm kiến thức về các thuật toán của lý thuyết đồ thị, hiểu biết khái quát hơn những ứng dụng thú vị và thực tế của môn học này.

## **BÀI BÁO CÁO CUỐI KỲ NÀY ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Em xin cam đoan đây là sản phẩm của riêng em được sự hướng dẫn của thầy Bùi Danh Hường. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính em thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong bài báo cáo này 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 bài báo cáo của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do em gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 19 tháng 05 năm 2020*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Tô Vĩnh Khang*

## **PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN**

### **Phần xác nhận của GV hướng dẫn**

---

---

---

---

---

---

---

Tp. Hồ Chí Minh, ngày      tháng      năm  
(ký và ghi họ tên)

### **Phần đánh giá của GV chấm bài**

---

---

---

---

---

---

---

Tp. Hồ Chí Minh, ngày      tháng      năm  
(ký và ghi họ tên)

## TÓM TẮT

Ngày nay, công nghệ thông tin đã và đang đóng vai trò quan trọng trong xã hội. Mọi người liên lạc, trao đổi thông tin với nhau chỉ qua một nơi trung gian gọi là mạng xã hội. Với quá trình phát triển không ngừng của công nghệ, mạng xã hội đã chiếm gần như tuyệt đối về nhu cầu người dùng. Và mô hình mạng xã hội được xây dựng dựa trên những ứng dụng của lý thuyết đồ thị.

Bài báo cáo này sẽ phân tích về mạng xã hội và các giải pháp hiện có bằng cách mô hình hóa bằng các biểu đồ, một vài phân tích thông qua các hệ số về mật độ, tính trung tâm bằng các công thức đã được đúc kết một cách đơn giản dễ hiểu và dễ sử dụng cũng như cách phân tích một vấn đề cụ thể để nhận diện người chủ chốt trong mạng xã hội. Bên cạnh đó, một số thuật toán phân cụm cũng sẽ được đề cập đến trong việc nhận diện cộng đồng và một số ứng dụng thực tế ngày nay.

## MỤC LỤC

MỤC LỤC.....	1
DANH MỤC CÁC HÌNH.....	3
CHƯƠNG 1 – MẠNG XÃ HỘI VÀ CÁC VẤN ĐỀ PHÂN TÍCH MẠNG XÃ HỘI....	4
1.1 Mạng xã hội.....	4
1.1.1 Khái niệm.....	4
1.1.2 Tầm quan trọng.....	5
1.1.3 Cách mô hình hóa bằng biểu đồ.....	5
1.2 Các vấn đề phân tích mạng xã hội.....	8
CHƯƠNG 2 – ĐO LƯỜNG MẠNG XÃ HỘI.....	10
2.1 Các kỹ thuật đo lường.....	10
2.1.1 Mật độ (Density).....	10
2.1.2 Tính trung tâm (Centrality).....	11
2.1.2.1 Trung tâm độ (Degree centrality).....	12
2.1.2.2 Trung tâm lân cận (Closeness centrality).....	13
2.1.2.3 Trung tâm trung gian (Betweenness centrality).....	14
2.1.2.4 Trung tâm phân cụm (Clustering centrality).....	15
2.2 Người chủ chốt và nhận diện người chủ chốt (Detect key players).....	16
2.3 Ứng dụng.....	18
CHƯƠNG 3 – NHẬN DIỆN CỘNG ĐỒNG.....	20
3.1 Hệ số Mô-đun (Modularity).....	20
3.2 Thuật toán Grivan Newman (GNA).....	22
3.3 Thuật toán Dựa trên nút tương tự (NSBA).....	23
3.4 Thuật toán Truyền nhãn (LPA).....	24
CHƯƠNG 4 – DEMO.....	26

4.1 Demo về 5 công thức thống kê đo lường mạng xã hội phổ biến.....	26
4.1.1 Demo về Mật độ (Density).....	28
4.1.2 Demo về trung tâm độ (Degree centrality).....	28
4.1.3 Demo về Trung tâm lân cận (Closeness centrality).....	31
4.1.4 Demo về Trung tâm trung gian (Betweenness centrality).....	33
4.1.5 Demo về Trung tâm phân cụm (Clustering centrality).....	35
4.2 Demo về một thuật toán nhận diện cộng đồng.....	38
CHƯƠNG 5 – TỔNG KẾT.....	43
TÀI LIỆU THAM KHẢO.....	44
TỰ ĐÁNH GIÁ.....	45



## DANH MỤC CÁC HÌNH

Hình 1.1 Hình ảnh về các mối liên hệ của một mạng xã hội.....	4
Hình 1.2 Hình ảnh về một biểu đồ liên kết với 6 nút và 5 đường.....	5
Hình 1.3 Hình ảnh về sự phân rã mối liên kết trong mạng khi loại bỏ điểm D.....	6
Hình 1.4 Hình ảnh về biểu đồ liên kết với 9 nút và 8 đường.....	7
Hình 1.5 Hình ảnh về sự phân cụm của biểu đồ thành 2 cụm nhỏ hơn.....	7
Hình 1.6 Hình ảnh về biểu đồ khi loại bỏ điểm D.....	8
Hình 2.1 Hình ảnh về một mạng nhân tạo với 11 nút và 10 cạnh.....	10
Hình 2.2 Hình ảnh về đồ thị mật độ cao và mật độ thấp.....	11
Hình 2.3 Đồ thị minh họa về trung tâm độ.....	12
Hình 2.4 Đồ thị minh họa về trung tâm lân cận.....	13
Hình 2.5 Đồ thị minh họa về trung tâm trung gian.....	14
Hình 2.6 Đồ thị minh họa về sự phân cụm của 2 mạng xã hội.....	15
Hình 2.7 Đồ thị minh họa về mối liên hệ của nhóm nhân viên phát triển phần mềm....	16
Hình 2.8 Đồ thị minh họa về nhận diện những người chủ chốt.....	17
Hình 2.9 Biểu đồ thống kê số lượng người dùng trong năm mạng xã hội.....	19
Hình 3.1 Một số ví dụ về tính toán phân chia cộng đồng bằng hệ số mô-đun.....	21
Hình 3.2 Hình ảnh về nhận diện cộng đồng bằng thuật toán Girvan-Newman.....	22
Hình 3.3 Hình ảnh về quá trình truyền nhãn bằng thuật toán LPA.....	25
Hình 4.1 Hình ảnh demo một mạng xã hội nhân tạo với 14 nút 16 cạnh.....	27
Hình 4.2 Hình ảnh mô phỏng cho việc demo thuật toán Girvan Newman.....	38
Hình 4.3 Kết quả phân cụm nhận diện cộng đồng (Câu lạc bộ karate Zachary).....	39
Hình 4.4 Kết quả phân cụm nhận diện cộng đồng (Mạng nhân tạo hình 4.1).....	42

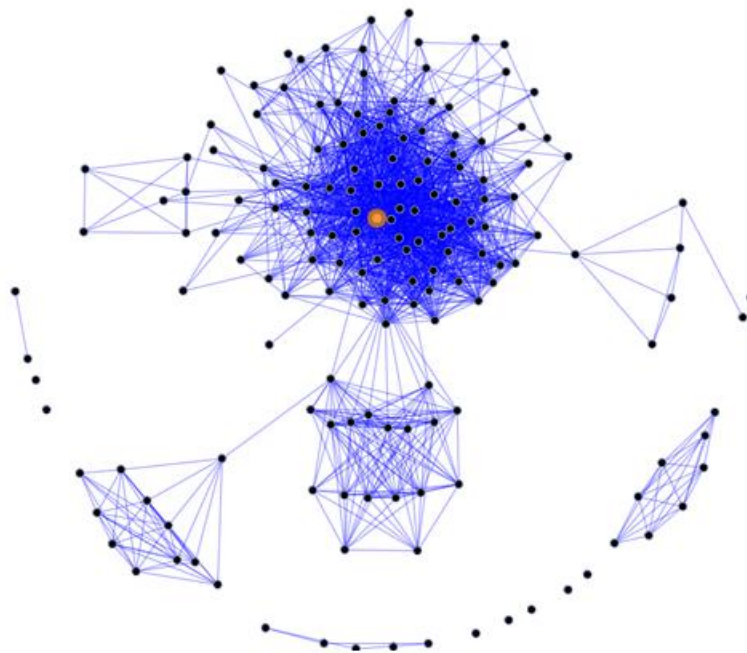
# CHƯƠNG 1 – MẠNG XÃ HỘI VÀ CÁC VẤN ĐỀ PHÂN TÍCH MẠNG XÃ HỘI

## 1.1 Mạng xã hội

### 1.1.1 Khái niệm

Mạng xã hội là một cấu trúc xã hội được tạo thành từ các cá nhân hoặc tổ chức được gọi là "nút", được kết nối bởi một hoặc nhiều loại phụ thuộc lẫn nhau. Các nút là các tác nhân riêng lẻ trong các mạng và các mối quan hệ là mối quan hệ giữa các diễn viên. Các cấu trúc dựa trên đồ thị kết quả thường rất phức tạp. Có thể có nhiều các loại quan hệ giữa các nút.

Ở dạng đơn giản nhất, mạng xã hội là bản đồ của các mối quan hệ được chỉ định, chẳng hạn như tình bạn, giữa các nút được nghiên cứu. Trong sơ đồ mạng xã hội, các nút là các điểm và các mối quan hệ là các đường.



**Hình 1.1 Hình ảnh về các mối liên hệ của một mạng xã hội**

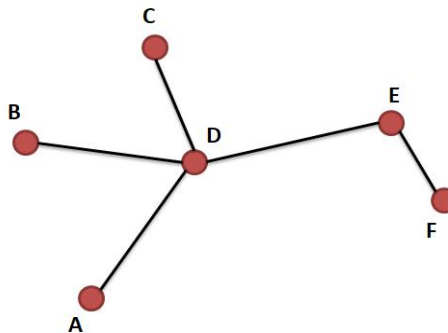
### 1.1.2 Tầm quan trọng

Để hiểu rõ một mạng xã hội, ta cần phải phân tích mạng xã hội đó. Thông qua quá trình phân tích, ta sẽ biết được những ưu khuyết điểm, biết được đâu là điểm nút của nó. Từ đó có thể khai thác các thông tin các giá trị, để phục vụ cho một nhu cầu nào đó. Phân tích mạng xã hội đã nổi lên như một kỹ thuật quan trọng trong xã hội học hiện đại. Nó cũng đã đạt được một số ý nghĩa trong nhân học, sinh học, nghiên cứu truyền thông, kinh tế, địa lý, khoa học thông tin, nghiên cứu tổ chức, tâm lý xã hội, và xã hội học, và đã trở thành một chủ đề phổ biến của đầu cơ và nghiên cứu. Hiểu rõ mạng lưới hoạt động của từng mạng xã hội, ta có thể tác động vào nó để giảm thiểu rủi ro, nắm bắt các thông tin về sở thích, hành vi con người...

### 1.1.3 Cách mô hình hóa bằng biểu đồ

*Phương pháp này đã được James Coleman thử nghiệm và triển khai (1964).*

Khái niệm toán học của đồ thị liên quan đến ý tưởng về các điểm nút được kết nối bởi các đường và lý thuyết đồ thị bao gồm một tập hợp các quy trình để phân tích sự hiện diện, hướng và độ mạnh của các đường nối các điểm nút này.

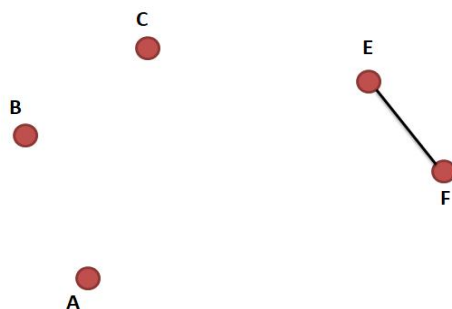


**Hình 1.2 Hình ảnh về một biểu đồ liên kết với 6 nút và 5 đường**

Biểu đồ liên kết đơn giản (Hình 1.2) gồm có 6 điểm: A,B,C,D,E,F. Được nối với nhau bằng 5 đường. Sự sắp xếp các điểm trên trang hoàn toàn tùy ý và có thể thay đổi

tùy ý khi phân tích. Mô hình biểu đồ liên kết là điều nhất thiết phải làm khi phân tích bằng phương pháp này. Việc phân tích sẽ càng dễ dàng khi nhà phân tích có thể sắp xếp các điểm nút sao cho rõ ràng nhất. Ở đây tôi đặt điểm C và B xuất hiện gần nhau hơn so với điểm C và A mà chưa cần quan tâm về khoảng cách vật lý. Giữa các điểm chỉ tính theo số lượng đường mà cần phải đi qua để đi từ điểm nút này đến điểm nút khác.

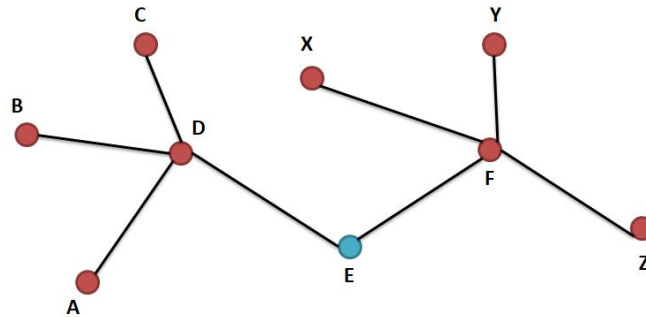
Khi quan sát biểu đồ, thấy rằng điểm D chính là điểm liên kết với các điểm khác nhiều nhất (4 điểm), điểm E liên kết với 2 điểm và các điểm A,B,C,F chỉ 1 điểm. Qua đó có thể thấy được mức độ quan trọng của điểm D trong biểu đồ do nó có nhiều mối liên kết nhất. Giả sử nếu loại bỏ điểm D sẽ thấy ngay việc phân rã gần như hoàn toàn các mối liên kết trong mạng xã hội. Ta gọi điểm D là điểm có mật độ cao nhất.



**Hình 1.3 Hình ảnh về sự phân rã mối liên kết trong mạng khi loại bỏ điểm D**

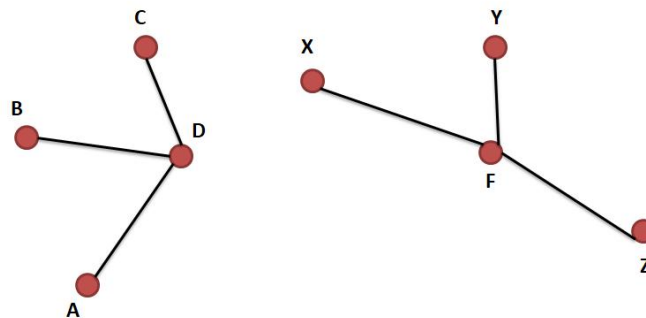
Tuy nhiên, điều này sẽ không còn hợp lý khi ta đặt ra giả thiết như sau. Chẳng hạn như một tập đoàn công nghệ có 1000 lập trình viên. Khi đó, các lập trình viên này có thể sẽ biết được nhau thông qua thẻ nhân viên. Nhưng không thể chắc chắn rằng tất cả các lập trình viên này sẽ vui vẻ khi làm việc cùng nhau. Mật độ tối đa cho mỗi quan hệ khi đó có thể thấp hơn nhiều. Do đó, việc đánh giá mật độ thực tế trong một mạng xã hội phải tính đến kích thước của mạng xã hội lúc đó. Các tính chất toán học về thước đo mật độ không được làm sai lệch trong quá trình phân tích (Barnes 1979). Một cụm có thể được xem như một điểm trong một biểu đồ lớn hơn và ít mật độ

hơn. Giả sử điểm F cũng có mối liên kết của riêng nó (ví dụ như X,Y,Z) thì biểu đồ tổng thể lúc này là bao gồm 2 cụm  $\{\{A,B,C,D\}, \{F,X,Y,Z\}\}$  và được liên kết thông qua điểm chung là E. Lúc này, điểm E lại là nơi có số bước thông qua nó nhiều nhất. Ta gọi điểm E là điểm có tính trung tâm trung gian cao nhất.



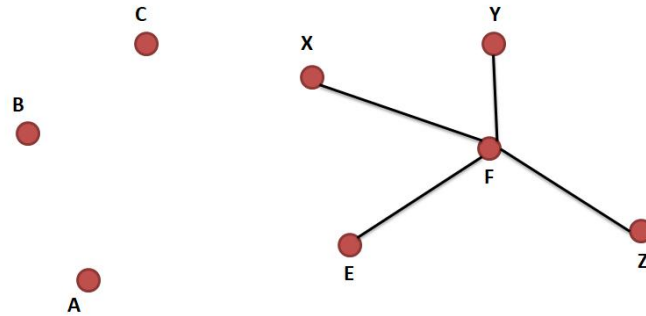
**Hình 1.4 Hình ảnh về biểu đồ liên kết với 9 nút và 8 đường**

Giả sử, ta loại bỏ điểm có tính trung tâm trung gian cao nhất - điểm E. Thì ta được một biểu đồ phân cụm rõ ràng như sau.



**Hình 1.5 Hình ảnh về sự phân cụm của biểu đồ thành 2 cụm nhỏ hơn**

Hoặc giả sử ta loại bỏ điểm ban đầu được xem là điểm có mật độ cao nhất - điểm D. Lúc này nếu loại bỏ điểm D thì mạng xã hội được xem là bị ảnh hưởng hơn. Do vẫn còn tồn tại điểm F với mật độ cao.



**Hình 1.6 Hình ảnh về biểu đồ khi loại bỏ điểm D**

Ngoài ra , vẫn còn những cách xác định khác thông qua mô hình hóa biểu đồ được các nhà phân tích lựa chọn như độ trung tâm , trung tâm lân cận ,..

## 1.2 Các vấn đề phân tích mạng xã hội

Phân tích mạng xã hội được sử dụng để đo lường và phân tích các thuộc tính cấu trúc của mạng mỗi quan hệ “dyadic” phụ thuộc lẫn nhau. Trong xã hội học, một “dyad” là một nhóm gồm hai người, nhóm xã hội nhỏ nhất có thể và "dyadic" là một tính từ mô tả sự tương tác của họ. Mô hình và cấu trúc của các mối quan hệ xã hội cung cấp các ràng buộc và cơ hội cho cá nhân tham gia và ảnh hưởng đến thế giới xã hội của họ. (Simmel 1950 Frisby và Sayer, 1986). Chính những ý tưởng này, Moreno đã được mở rộng và vận dụng lý thuyết đồ thị và đại số quan hệ để tạo đồ thị và phân tích mạng xã hội đầu tiên (1953). Tiến bộ trong việc phân tích mạng xã hội đã tạo ra một loạt các khái niệm, giải pháp và kỹ thuật.

Các giải pháp cấp độ nút được sử dụng để mô tả và kiểm tra các nút của một mạng. Đây là đặc điểm của các nút có thể được phân lớp vào phân tích dưới dạng các giải pháp cấp nút. Một giải pháp cấp nút khác, và có lẽ là giải pháp chính đó là tính trung tâm. Tính trung tâm có thể được tính toán theo nhiều cách khác nhau và được sử dụng để xác định các nút chính hoặc nút trung tâm trong mạng. Tính trung tâm lân cận

đo lường tầm với của một nút đã cho hay đúng hơn là cách một nút đã cho gần với các nút khác một mạng lưới xã hội. Trong phân tích cấp nút , khoảng cách hoặc đường dẫn là thước đo số lượng quan hệ tối thiểu giữa hai nút hoặc mức độ phân tách giữa hai nút điểm giao. Khoảng cách là nền tảng cho tất cả các biện pháp cấp nút và có thể được sử dụng ở cấp độ phân tích mạng để mô tả kích thước tổng thể của một mạng.

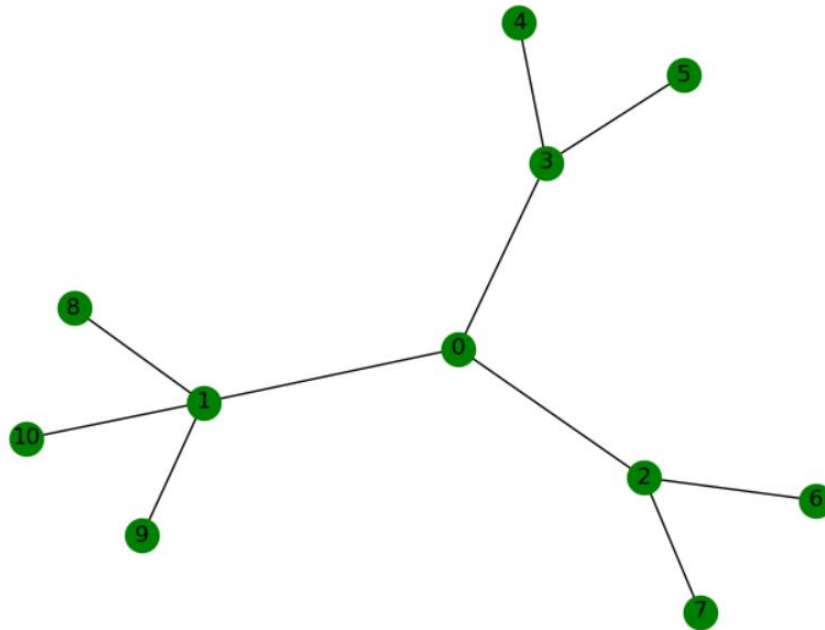
Các giải pháp cấp mạng được sử dụng để mô tả và kiểm tra cấu trúc tổng thể của một mạng xã hội và thường gọi là các giải pháp gắn kết mạng. Mật độ là giải pháp cấp mạng được sử dụng nhiều nhất. Mật độ là tỷ lệ của số lượng quan hệ thực tế trong một mạng với số lượng quan hệ có thể trong một mạng. Mật độ càng cao có nghĩa là kết nối nhiều hơn giữa các nút trong mạng. Khoảng cách, có thể được sử dụng ở cấp độ mạng sao cho khoảng cách lớn nhất trong một mạng đại diện cho đường kính của nó và khoảng cách trung bình trong một mạng lưới cho chúng ta một ý nghĩa tổng thể về cách tách biệt nhất các nút là từ nhau. Tương tự, trung tâm có thể được tính cho một mạng để xác định mức độ tập trung một mạng là, xác định nếu hầu hết các mối quan hệ thuộc về một tập hợp nhỏ các nút hoặc được chia đều hơn trên các nút. Ngoài ra, việc phân tích phân cụm có thể được tiến hành ở cấp độ mạng để xác định xem có các cụm nút có ý nghĩa được phân tách không trong một mạng, giống như tính lân cận, nó có thể liên quan đến các luồng trong các mạng được định hướng.

Có rất nhiều giải pháp kỹ thuật phân tích mạng xã hội như kích thước (số nút trong mạng) , chiều dài (số cạnh liên kết trong mạng) , bán kính (khoảng cách tối thiểu giữa 2 nút trong mạng) , đường kính (khoảng cách tối đa giữa 2 nút trong mạng) , độ lệch (cách phân phối độ nút) , ... Tuy nhiên , ở bài báo cáo này sẽ chỉ tập trung phân tích về một số hệ số đo lường mạng xã hội thông dụng và nhận diện cộng đồng. Chi tiết cụ thể về việc phân tích được trình bày ở chương 2 và 3.

## CHƯƠNG 2 – ĐO LƯỜNG MẠNG XÃ HỘI

### 2.1 Các kỹ thuật đo lường

Dựa vào các giải pháp phân tích xã hội đã hình thành rất nhiều kỹ thuật. Các kỹ thuật đo lường được sử dụng trong phân tích mạng xã hội dựa trên nguyên lý của lý thuyết đồ thị, bao gồm một tập hợp các công thức toán học và khái niệm cho việc nghiên cứu các mẫu của dòng. Dưới đây là một trong những hệ số đo lường thông dụng nhất khi phân tích mạng xã hội.

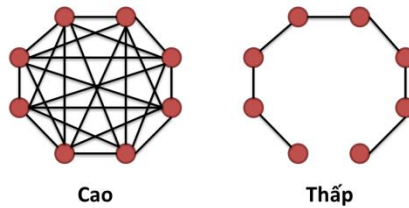


**Hình 2.1 Hình ảnh về một mạng nhân tạo với 11 nút và 10 cạnh**

#### 2.1.1 Mật độ (Density)

- Ý nghĩa: Dùng để so sánh sự mạnh yếu của hai hay nhiều mạng lưới với nhau. Khi hệ số mật độ của mạng xã hội càng lớn nghĩa là mức độ gắn kết, sự chặt chẽ của các mối quan hệ giữa các nút trong mạng lưới cũng càng lớn. Nó cho ta biết được sự tương hỗ giữa các nút càng nhiều, càng hiệu quả, sự điều tiết của mạng xã hội đối hành vi của nút cũng càng mạnh mẽ hơn và ngược lại.





**Hình 2.2 Hình ảnh về đồ thị mật độ cao và mật độ thấp**

- Công thức:

$$D = \frac{m}{n(n-1)/2} \quad (0 \leq D \leq 1)$$

Với:

- m là Tổng số các mối liên hệ thực tế của mạng xã hội
- n là Tổng số các nút trong mạng xã hội

- Ví dụ: Giả sử ta có một mạng xã hội như Hình 2.1, áp dụng công thức ta tìm được mật độ của mạng xã hội nhân tạo này là:

$$D = \frac{10}{11(11-1)/2} \approx 0.18$$

Với mật độ là 0.18 , ta dễ dàng nhận thấy mạng xã hội này thưa thớt, kém chặt chẽ.

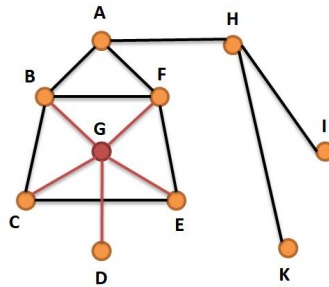
### **2.1.2 Tính trung tâm (Centrality)**

Trong lý thuyết đồ thị và phân tích mạng, có nhiều thước đo khác nhau về tính trung tâm của một đỉnh trong đồ thị xác định tầm quan trọng tương đối của một đỉnh trong biểu đồ (ví dụ: mức độ quan trọng của một người quản lý trong một công ty hoặc theo lý thuyết về cú pháp không gian, một căn phòng quan trọng như thế nào trong một tòa nhà hoặc một con đường được sử dụng tốt như thế nào là trong một mạng lưới đô thị).

Có bốn thước đo tính trung tâm được sử dụng rộng rãi trong phân tích mạng: trung tâm độ, trung tâm lân cận, trung tâm trung gian và trung tâm phân cụm.

### 2.1.2.1 Trung tâm độ (Degree centrality)

• Ý nghĩa: Dùng để đo lường được số lượng của các mối quan hệ trực tiếp của một nút nào đó với các nút khác trong mạng xã hội. Khi hệ số trung tâm độ của mạng xã hội càng lớn nghĩa là càng nằm ở vị trí trung tâm của mạng lưới xã hội. Khi đó, ưu thế của nút đó trong mạng lưới được thể hiện rõ, càng ít bị phụ thuộc cũng như càng có nhiều cơ hội khai thác, thao túng được các nút khác trong mạng xã hội. Tuy nhiên, khuyết điểm của hệ số trung tâm độ là chỉ tính các mối quan hệ trực tiếp của nút, chưa thể chắc chắn nút đó có hệ số trung tâm độ lớn là chiếm toàn bộ ưu thế so với các nút khác.



Hình 2.3 Đồ thị minh họa về trung tâm độ

• Công thức:

$$C_d = \frac{m}{n-1} \quad (0 \leq C_d \leq 1)$$

Với:

- m là Tổng số các mối quan hệ trực tiếp của nút

- n là Tổng số các nút trong mạng xã hội

• Ví dụ: Theo Hình 2.1, ta sẽ tính toán lần lượt các hệ số trung tâm độ của từng nút trong mạng:

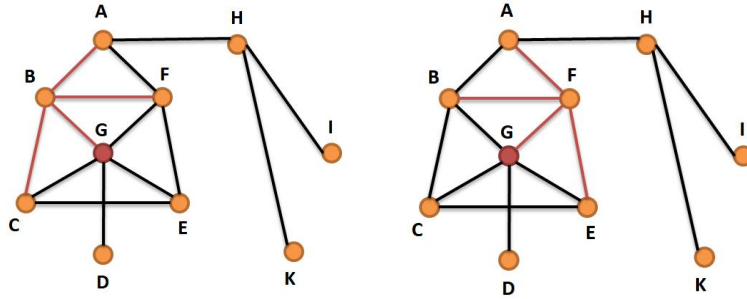
$$C_{d(\text{nút0})} = C_{d(\text{nút2})} = C_{d(\text{nút3})} = \frac{3}{11-1} = 0.3 ; C_{d(\text{nút1})} = \frac{4}{11-1} = 0.4$$

$$C_{d(\text{nút4})} = C_{d(\text{nút5})} = C_{d(\text{nút6})} = C_{d(\text{nút7})} = C_{d(\text{nút8})} = C_{d(\text{nút9})} = C_{d(\text{nút10})} = \frac{1}{11-1} = 0.1$$

Với nút 1 có hệ số trung tâm độ lớn nhất (0.4) nên nó có tầm ảnh hưởng rất lớn đến mạng xã hội này (Hình 2.1).

### 2.1.2.2 Trung tâm lân cận (Closeness centrality)

- Ý nghĩa: Dùng để hỗ trợ cho việc xác định ưu thế của nút sau khi tính toán hệ số trung tâm độ. Khi hệ số trung tâm lân cận của mạng xã hội càng lớn nghĩa là tính trung tâm càng cao, khoảng cách từ nút đó đến các nút khác càng ngắn, càng hỗ trợ cho nút đó chiếm ưu thế và ảnh hưởng lên toàn bộ mạng xã hội.



Hình 2.4 Đồ thị minh họa về trung tâm lân cận

- Công thức:

$$C_c = \frac{n-1}{s} \quad (0 \leq C_c \leq 1)$$

Với:

- s là Tổng số bước của đoạn đường ngắn nhất từ nút được xác định đến các nút còn lại trong mạng xã hội.
- n là Tổng số các nút trong mạng xã hội.

- Ví dụ: Tiếp tục với (Hình 2.1), ta tính lần lượt các hệ số trung tâm lân cận của từng nút trong mạng và được các kết quả như sau:

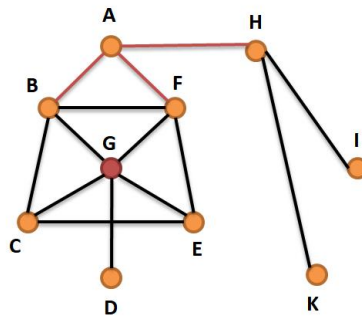
$$C_{c(\text{nút}0)} \approx 0.58 ; C_{c(\text{nút}1)} = 0.5 ; C_{c(\text{nút}2)} = C_{c(\text{nút}3)} \approx 0.45$$

$$C_{c(\text{nút}4)} = C_{c(\text{nút}5)} = C_{c(\text{nút}6)} = C_{c(\text{nút}7)} \approx 0.32 ; C_{c(\text{nút}8)} = C_{c(\text{nút}9)} = C_{c(\text{nút}10)} \approx 0.35$$

Với nút 0 cao nhất, vì bản chất nó là nút trung tâm độ, do đó xét các nút lân cận thấy rằng nút 1,2,3 là các nút có trung tâm lân cận sát với nút 0 và hỗ trợ cho nút 0 trong mạng.

### 2.1.2.3 Trung tâm trung gian (Betweenness centrality)

• Ý nghĩa: Dùng để đo lường một nút đóng vai trò trung gian trong việc trao đổi thông tin với các nút khác trong mạng xã hội. Khi hệ số trung tâm trung gian càng lớn nghĩa là nút này được các nút khác trao đổi thông tin đến và đi nhiều nhất, nắm giữ nhiều thông tin trung gian. Do đó, nút này sẽ thuận lợi trong việc kiểm soát mọi thông tin trong mạng lưới và có thể tác động đến mạng lưới một cách dễ dàng theo hướng có lợi cho nó nếu muốn. Nhưng đồng thời nút này cũng đứng ở vị trí tốt nhất để thúc đẩy sự phối hợp giữa các nút khác trong mạng lưới.



**Hình 2.5 Đồ thị minh họa về trung tâm trung gian**

• Công thức:

$$C_b = \frac{k}{(n-1)(n-2)/2} \quad (0 \leq C_b \leq 1)$$

Với:

- k là Tổng số lần làm trung gian cho các nút khác trong mạng xã hội
- n là Tổng số các nút trong mạng xã hội.

• Ví dụ: Vẫn là (Hình 2.1), ta tính lần lượt các hệ số trung tâm trung gian của từng nút trong mạng và được các kết quả như sau:

$$C_{b(\text{nút0})} \approx 0.73 ; C_{b(\text{nút1})} \approx 0.53 ; C_{b(\text{nút2})} = C_{b(\text{nút3})} \approx 0.37$$

$$C_{b(\text{nút4})} = C_{b(\text{nút5})} = C_{b(\text{nút6})} = C_{b(\text{nút7})} = C_{b(\text{nút8})} = C_{b(\text{nút9})} = C_{b(\text{nút10})} = 0$$

Nút 0 với số lượt làm trung gian nhiều nhất (0.73) và các nút từ 4 đến 10 nằm rìa nên hệ số trung tâm trung gian của nó là 0.

#### 2.1.2.4 Trung tâm phân cụm (Clustering centrality)

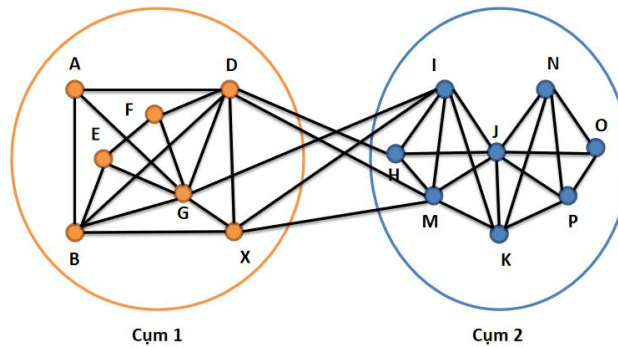
• Ý nghĩa: Dùng để biết số lượng bộ ba đóng trong vùng lân cận trên tổng số bộ ba trong vùng lân cận. Khi hệ số trung tâm phân cụm càng lớn nghĩa là vùng lân cận của một nút đó có kết nối càng chặt chẽ trong mạng xã hội.

• Công thức:

$$C_{\text{CLUSTERING}} = \frac{2m}{p(p-1)} \quad (0 \leq C_{\text{CLUSTERING}} \leq 1)$$

Với:

- m là Tổng số các mối quan hệ trực tiếp của nút được xác định
- p là Tổng số các liên kết trực tiếp (không qua trung gian) giữa hàng xóm của nút.



**Hình 2.6 Đồ thị minh họa về sự phân cụm của 2 mạng xã hội**

Hệ số trung tâm phân cụm của đồ thị là hệ số trung tâm phân cụm trung bình của tất cả các nút trong biểu đồ.

• Công thức:

$$\bar{C} = \frac{1}{n} \sum_{i=1}^n C_{\text{CLUSTERING}(i)} \quad (0 \leq \bar{C} \leq 1)$$

Với: n là Tổng số các  $C_{\text{CLUSTERING}}$  đã tính được.

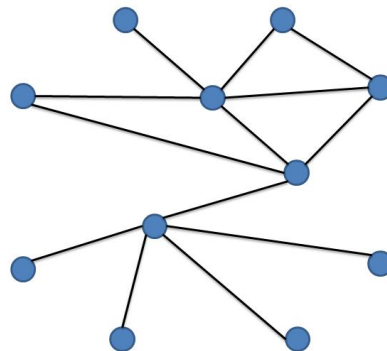
• Ví dụ: Dựa vào (Hình 2.1), ta có thể thấy do mạng này không phân cụm nên chắc chắn các hệ số trung tâm phân cụm của từng nút lúc này sẽ đều bằng 0.

## 2.2 Người chủ chốt và nhận diện người chủ chốt (Detect key players)

Người chủ chốt là người (nút) có vai trò quan trọng nhất (theo nhu cầu của nhà phân tích) trong mạng xã hội. Tầm ảnh hưởng, các tác động của người này đến mạng là lớn nhất - là một mắt xích quan trọng. Ở đây, người chủ chốt có thể là một hoặc nhiều người bởi vì nhu cầu của nhà phân tích mạng xã hội là khác nhau. Đối với mỗi nhu cầu, tiêu chí, đánh giá riêng sẽ có những người chủ chốt riêng. Trước khi đi vào cách để phát hiện người mấu chốt này ta cần xem qua một ví dụ dưới đây để hiểu hơn về cách thức, nguyên lý hoạt động, các qui trình của một mạng xã hội:

- Giả sử:

Trong một công ty phát triển phần mềm (Hình 2.8). Sản phẩm phần mềm là một phần quan trọng trong việc thu lợi cho công ty. Nếu không có tạo ra sản phẩm thì công ty sẽ không thu được tiền, không có tiền sẽ dẫn đến người chủ công ty không thể phát lương cao hoặc thậm chí không thể phát lương cho các nhân viên của mình từ khâu quản lý dự án đến các lập trình viên.



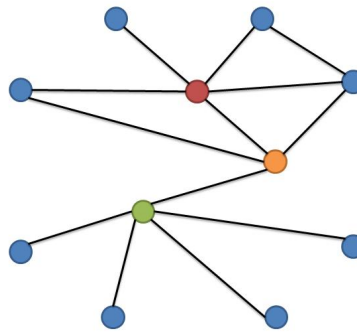
**Hình 2.7 Đồ thị minh họa về mối liên hệ của nhóm nhân viên phát triển phần mềm**

Người quản lý dự án (Project manager) nắm giữ thông tin của hầu hết các thành viên và các hoạt động, các thảo luận cho việc phát triển dự án phần mềm - người nắm giữ thông tin nhiều nhất. Vị trí của người này luôn là người am hiểu các thành viên (lập trình viên) của nhóm hoặc của một dự án. Nếu như một công ty phát triển mà mỗi lập

trình viên tự xây dựng mà không có bản thảo , không có sự chỉ dẫn , không có mối quan hệ liên kết , phân chia công việc cho các lập trình viên khác thì nguy cơ tạo ra vô vàn sản phẩm bị lỗi , bị trùng lặp, tiêu tốn thời gian ,... Chính vì thế , người quản lý dự án là người giữ vai trò mấu chốt cho việc sản xuất ra các sản phẩm, chịu trách nhiệm chính trong việc đốc thúc quá trình tạo ra sản phẩm phần mềm cho công ty để công ty thu lợi nhuận.

Về phía người chủ công ty hoặc giám đốc (chức vụ tối cao), tuy là người nắm trọn quyền hành trong công ty , có thể sa thải bất kì người nào. Là người duy nhất có thể đưa ra quyết định về một hoặc tất cả hoạt động trong công ty. Thông thường các nhà lãnh đạo chỉ có mối quan hệ trong mạng lưới với người quản lý dự án và có rất ít mối quan hệ với các lập trình viên trong đó. Vì vậy, đến lượt người quản lý dự án sẽ tiếp xúc trực tiếp với những người liên quan khác.

- Nhận diện người chủ chốt (detect key players)



**Hình 2.8 Đồ thị minh họa về nhận diện những người chủ chốt**

Trong Hình 2.9 , nút màu đỏ là nút có hệ số trung tâm độ cao nhất. Nhưng nút cam và nút xanh lá cùng nhau sẽ đạt được nhiều nút hơn, sự ràng buộc của 2 nút này thông qua một liên kết ở giữa , nếu bị cắt đứt thì mạng sẽ chia thành 2 mạng con bị cô lập. Ứng với các màu đỏ , cam , xanh lá lần lượt là quản lý dự án , quản lý giao diện game ,

trưởng nhóm lập trình game. Người quản lý dự án có mối quan hệ với người quản lý về mảng giao diện cho game của công ty - am hiểu cách thức làm việc của nhóm lập trình game. Người quản lý giao diện game có mối quan hệ với người am hiểu các kỹ thuật lập trình, sử dụng ngôn ngữ nào cho phù hợp với việc triển khai loại game đó - trưởng nhóm lập trình game. Nếu như mối quan hệ giữa quản lý giao diện game và trưởng nhóm lập trình game bị chia cắt, hậu quả đó là khi người quản lý dự án yêu cầu triển khai loại game nào đó thì người quản lý giao diện game không có người triển khai thực hiện, phải mất thời gian tìm người thay thế. Còn người trưởng nhóm lập trình game lại chỉ quanh quẩn với 4 thành viên còn lại mà không biết phải làm game gì tiếp theo. Hệ lụy dẫn đến đó là quản lý dự án chậm trễ trong việc phát hành game; Chủ công ty không thu được lợi nhuận. Đây chỉ là một ví dụ minh họa về sự ảnh hưởng, các tác động của những người chủ chốt trong mạng xã hội trong số vô vàn tình huống.

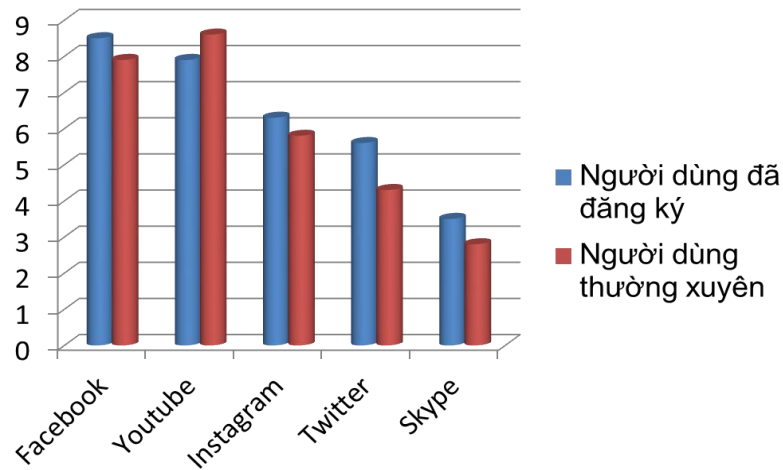
Vì vậy, để xác định được những người chủ chốt (key players) trong mạng xã hội và tra hỏi tiến trình tạo ra sản phẩm của công ty trước tiên phải trải qua quá trình thống kê các mối quan hệ (mối liên kết) trong công ty (mạng). Tính toán và so sánh bằng các phương pháp được nêu phía trên phù hợp với nhu cầu tìm kiếm của mình.

## 2.3 Ứng dụng

- Ứng dụng trong nghiên cứu: Mô hình ra quyết định, Mô hình tổ chức, Chuỗi cung ứng và mô hình quy trình, Hệ thống nhận thức và niềm tin của con người,...
- Ý nghĩa của việc ứng dụng phân tích mạng xã hội:
  - Phân tích để xác định các thuộc tính cấu trúc của một mạng (Ví dụ: phân phối độ đỉnh và đường kính của đồ thị).
  - Phân tích để tìm một số lượng có thể đo được trong mạng (Ví dụ: đối với mạng giao thông, mức độ dòng chảy trong bất kỳ phân nào của nó).
  - Phân tích được các tính chất động của mạng. (Ví dụ: Sự thay đổi của một mạng xã hội nào đó sau 10 năm khi nhu cầu, sở thích của người dùng thay đổi).



- Một vài sản phẩm được ứng dụng phân tích mạng xã hội:



**Hình 2.9 Biểu đồ thống kê số lượng người dùng trong năm mạng xã hội**

(Nguồn: *GlobalWebIndex Flagship Report 2018 - Không tính Trung Quốc*)

Facebook, Inc. là một công ty truyền thông xã hội và công nghệ Mỹ có trụ sở tại Menlo Park, California. Nó được thành lập bởi Mark Zuckerberg, cùng với các sinh viên và bạn cùng phòng của Đại học Harvard. Theo thống kê của GlobalWebIndex (2018) thì ứng dụng Facebook hiện dẫn đầu thế giới với 2,23 tỉ người dùng, có đến 65 triệu doanh nghiệp lập trang thông tin trên mạng này.

Nguyên nhân chính để Facebook thu hút được một số lượng cực lớn như hiện nay là do công ty này luôn có những đội ngũ là các nhà phân tích mạng xã hội. Họ sẽ thu thập thông tin người dùng, các nhu cầu sở thích, các hành vi hằng ngày khi truy cập ứng dụng,... Từ đó, thống kê lại và tìm ra những “key players”, phát hiện cộng đồng mới trong cộng đồng hiện có và phát triển nó cho phù hợp hơn để kéo lượng người dùng đến sử dụng cũng như “giữ chân” những người đã sử dụng qua ứng dụng Facebook. Dần dần, quá trình xây dựng đó đã khiến Facebook có được thành công như ngày hôm nay. Cách để phát hiện cộng đồng, phân cụm chúng là những gì sẽ được nói đến qua chương tiếp theo.

## CHƯƠNG 3 – NHẬN DIỆN CỘNG ĐỒNG

### 3.1 Hệ số Mô-đun (Modularity)

Là một thước đo cấu trúc của một mạng xã hội hoặc một đồ thị. Nó được thiết kế để đo lường việc phân chia một mạng xã hội thành các mô-đun (còn được gọi là cụm hoặc cộng đồng). Nó phản ánh mức độ tập trung của các cạnh trong các mô-đun so với phân phối ngẫu nhiên các liên kết giữa tất cả các nút bất kể mô-đun.

Các mạng xã hội có tính mô-đun cao có các kết nối dày đặc giữa các nút trong các mô-đun nhưng kết nối thưa thớt giữa các nút trong các mô-đun khác nhau. Tính mô-đun thường được sử dụng trong các phương pháp tối ưu hóa để nhận diện cấu trúc cộng đồng.

Giả sử một mạng lưới có  $n$  đỉnh liên kết với nhau bởi  $m$  cạnh. Đặt  $A_{ij}$  là một phần tử của ma trận kề của đồ thị, nghĩa là  $A_{ij}$  là số lượng cạnh giữa đỉnh  $i$  và  $j$ . Giả sử có một phép chia các đỉnh thành  $c$  cụm, cần tính giá trị hệ mô-đun cho phép chia này. Khi đó, với mỗi cặp đỉnh  $i$  và  $j$ , ta tính bằng công thức sau:

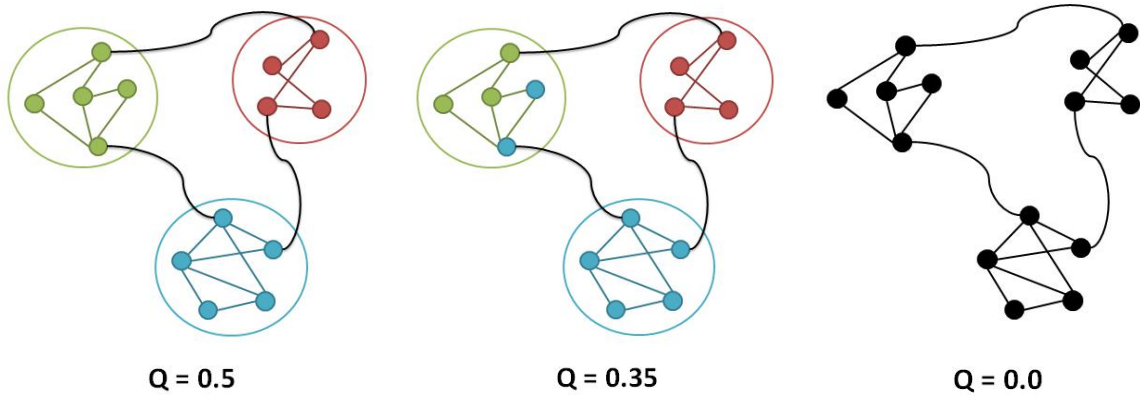
$$Q = \frac{1}{2m} \sum_{i,j} \left[ A_{ij} - \frac{k_i k_j}{2m} \right] \delta(C_i, C_j)$$

Hàm  $\delta$  có giá trị bằng 1 nếu  $i, j$  thuộc cùng một cộng đồng và bằng 0 nếu ngược lại (Newman 2006). Phương pháp tối ưu hóa cho thuật toán này cũng được triển khai và dễ hiểu hơn bằng cách phân chia mạng sau đó tính giá trị của mô-đun:

$$Q = \sum_{c=1}^{n_c} \left[ \frac{a_c}{m} - \left( \frac{b_c}{2m} \right)^2 \right]$$

Với:

- $a_c$  là Tổng số các liên kết giữa các nút trong cộng đồng
- $b_c$  là Tổng các độ của các nút trong cộng đồng
- $m$  là Tổng số liên kết.



**Hình 3.1 Một số ví dụ về tính toán phân chia cộng đồng bằng hệ số mô-đun**

Giả sử mạng xã hội lúc này được phân chia thành 3 cộng đồng (cụm), việc tính toán hệ số mô-đun để biết được mức độ phân cụm rõ ràng hay không. Đối với Hình 3.1, Trường hợp đầu tiên các cộng đồng được phân chia rõ ràng (chẳng hạn như có điểm chung về sở thích ăn uống, du lịch,... sẽ đưa về cùng một cộng đồng; nghiên cứu khoa học sẽ đưa về cùng một cộng đồng,...).

Hãy nhớ lại công thức tối ưu hóa phía trên. Trong trường hợp đầu tiên, cách tính toán phân cụm là cụm xanh lá ( $Q_{\text{xanhlá}} = [6 \text{ liên kết trong cộng đồng} / 20 \text{ liên kết trong mạng}] - (14 \text{ nút độ trong mạng} / 2 * 20 \text{ liên kết trong mạng}) = 0.1775$ ). Tương tự cho cụm đỏ  $Q_{\text{đỏ}} = 0.1375$ , cụm xanh dương  $Q_{\text{xanhdương}} = 0.19$ . Hệ số mô-đun phân cụm của mạng xã hội lúc này sẽ là  $Q = Q_{\text{xanhlá}} + Q_{\text{đỏ}} + Q_{\text{xanhdương}} = 0.1775 + 0.1375 + 0.19 = 0.505 \approx 0.5$ . Trường hợp 2, do bị tác động một phần (cụm xanh lá chứa 2 nút xanh dương) nên có sự thay đổi rõ rệt về hệ số mô-đun lúc này,  $Q = 0.06 + 0.1475 + 0.1375 = 0.345 \approx 0.35$ . Ở trường hợp cuối cùng, do mạng không phân cụm nên hệ số mô-đun lúc này là không,  $Q = 0$ .

Qua những ví dụ trên có thể thấy hệ số mô-đun  $Q$  càng lớn có nghĩa là kết quả nhận diện cộng đồng của một mạng lưới xã hội càng cao.

### 3.2 Thuật toán Grivan Newman (GNA)

Thuật toán Grivan Newman được áp dụng trong việc phân cụm bằng cách xác định cạnh có tính trung gian lớn nhất và loại bỏ nó. Một đường đi giữa 2 đỉnh trong mạng thuộc 2 cộng đồng khác nhau nhất thiết phải đi qua ít nhất một trong số các cạnh như vậy, vì thế nếu ta thiết lập các đường đi giữa tất cả các đỉnh trong mạng và xác định được cạnh nào trong đồ thị được các đường đi này đi qua nhiều nhất, ta có thể loại bỏ nó để phân chia mạng thành các cộng đồng riêng biệt.

Nếu trong trường hợp có 2 cạnh với độ trung gian như nhau, ta có thể loại bỏ một cạnh bất kỳ, hoặc loại bỏ cả 2 cạnh cùng một lúc.

Thuật toán Girvan Newman đưa lại kết quả tương đối tốt trong nhiều trường hợp. Tuy nhiên, nó tồn tại 2 nhược điểm. Một là, cũng như các thuật toán thuộc loại truyền thống, thuật toán Girvan-Newman không xác định trước được số cộng đồng mà mạng phân chia ra sẽ là bao nhiêu. Hai là, thuật toán Girvan và Newman không giải quyết được sự chồng chéo cộng đồng (một đỉnh có thể thuộc vào nhiều cộng đồng khác nhau trong mạng).

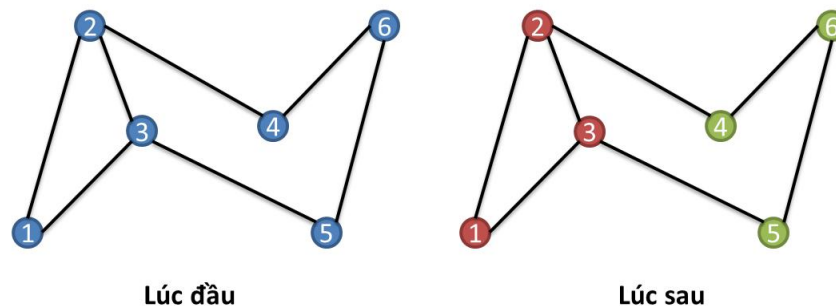
Thuật toán được thực hiện như sau:

*Bước 1: Tính khoảng cách giữa các cạnh cho mọi cạnh trong đồ thị.*

*Bước 2: Loại bỏ các cạnh có trung tâm trung gian cao nhất.*

*Bước 3: Tính khoảng cách giữa các cạnh cho các cạnh còn lại.*

*Bước 4: Lặp lại các bước 2 đến 4 cho đến khi tất cả các cạnh được loại bỏ.*



**Hình 3.2 Hình ảnh về nhận diện cộng đồng bằng thuật toán Girvan-Newman**

### 3.3 Thuật toán Dựa trên nút tương tự (NSBA)

Là một thuật toán dùng để chỉ ra rằng nếu hai nút giống nhau hơn được qui về nút tương tự nhau. Các nút có thể đại diện cho mọi người và các liên kết có thể đại diện cho tình bạn giữa các cá nhân như đã được nói ở phần đầu. Nếu hai người có cùng sở thích, hoàn cảnh hoặc bạn bè, họ có thể giống nhau. Hiện nay, có rất nhiều thuật toán dựa trên nút tương tự đã được nghiên cứu và chứng minh tính ưu việt cũng như những bất cập trong việc xét sự tương tự làm mất đi một số thông tin có thể xem là quan trọng của một nút nào đó.

Gọi  $N_i$  là vùng lân cận của đỉnh  $i$  trong một mạng (tập hợp các nút được kết nối trực tiếp với  $i$  thông qua một cạnh). Khi đó, số đo tương tự cho hai nút  $i$  và  $j$  được áp dụng bởi 3 thuật toán thông dụng sau:

+ Số đo Jacard (Paul Jacard):

$$\sigma(i, j) = J_{i,j} = \frac{|N_i \cap N_j|}{|N_i \cup N_j|}$$

+ Số đo Cosine (Gerard Salton):

$$\sigma(i, j) = C_{i,j} = \frac{|N_i \cap N_j|}{\sqrt{|N_i| \cdot |N_j|}}$$

+ Số đo Geometric (Goldberg-Roth):

$$\sigma(i, j) = G_{i,j} = \frac{|N_i \cap N_j|^2}{(|N_i| \cdot |N_j|)}$$

Lấy ví dụ như Hình 3.2 (Lúc đầu), chọn hai nút  $i, j$  lần lượt là 2 và 5 thì ta tính toán được các số đo tương ứng như sau:

$$\sigma(2,5) = J_{2,5} = \frac{|\{1,3,4\} \cap \{3,6\}|}{|\{1,3,4\} \cup \{3,6\}|} = \frac{1}{4} = 0.25$$

$$\sigma(2,5) = C_{2,5} = \frac{|\{1,3,4\} \cap \{3,6\}|}{\sqrt{|\{1,3,4\}| \cdot |\{3,6\}|}} = \frac{1}{\sqrt{6}} \approx 0.408$$

$$\sigma(2,5) = G_{2,5} = \frac{|\{1,3,4\} \cap \{3,6\}|^2}{(|\{1,3,4\}| \cdot |\{3,6\}|)} = \frac{1}{6} \approx 0.167$$

### 3.4 Thuật toán Truyền nhãn (LPA)

Là một thuật toán dùng để gán nhãn cho các điểm dữ liệu chưa được gán nhãn trước đó. Nhãn sẽ được truyền đến các điểm chưa được gán nhãn trong suốt quá trình của thuật toán. Việc truyền nhãn có lợi thế về thời gian hoạt động và lượng thông tin tiên nghiệm cần thiết về cấu trúc mạng hay có thể hiểu là không cần biết trước bất kì thông số nào. Tuy nhiên nó không phải là giải pháp duy nhất, mà là tổng hợp của nhiều giải pháp.

Thuật toán được thực hiện như sau:

-*Bước 1: Khởi tạo nhãn tại tất cả các nút trong mạng. Cho một nút  $x \Rightarrow C_x(0)=x$*

-*Bước 2: Đặt  $y = 1$*

-*Bước 3: Sắp xếp các nút theo thứ tự ngẫu nhiên và đặt chúng là tập hợp  $A$ .*

-*Bước 4: Với mỗi  $x \in A$  theo thứ tự đã sắp xếp. Cho:*

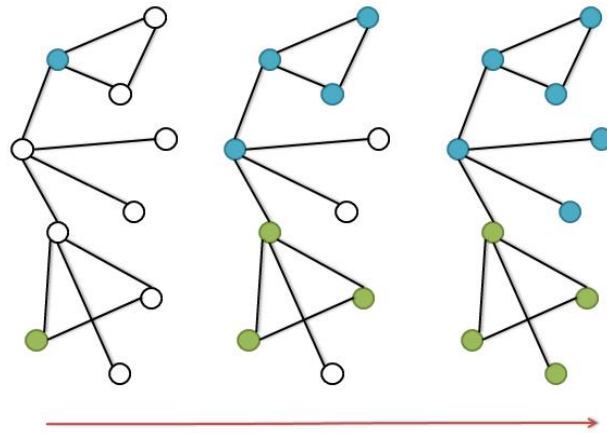
$$C_x(y) = f(C_{x_{i1}}(y), \dots, C_{x_{im}}(y), C_{x_{i(m+1)}}(y-1), \dots, C_{x_{ik}}(y-1)).$$

*Với  $f$  trả về nhãn xảy ra với tần suất cao nhất giữa các nút liên kết với nó và các mối liên kết bị phá vỡ đồng đều ngẫu nhiên.*

-*Bước 5:*

*+Nếu mỗi nút có nhãn là số lượng tối đa các nút liên kết với nó  $\Rightarrow$  dừng thuật toán.*

*+Nếu không, đặt  $y = y + 1 \Rightarrow$  quay về Bước 3.*



**Hình 3.3 Hình ảnh về quá trình truyền nhãn bằng thuật toán LPA**

Một trong những nhược điểm rõ ràng nhất của LPA là vấn đề lan truyền quá mức. Lý do chính đằng sau sự lan truyền quá mức là sự mở rộng nhanh chóng và tích cực của một số nút trong một số cộng đồng khác hay nói cách khác đó là cộng đồng yếu hơn có rất ít cơ hội phát triển và thậm chí bị nuốt chửng bởi các cộng đồng khác.

## CHƯƠNG 4 – DEMO

### 4.1 Demo về 5 công thức thống kê đo lường mạng xã hội phổ biến

- **Chú thích:**

-Đổi số truyền vào: Một mạng xã hội nhân tạo gồm 14 nút , 16 cạnh

***DataArtificialSocialNetwork.txt***

-Thực thi: chạy câu lệnh trong terminal tại thư mục ***SourceCode***

***python 51800408\_SocialNetworkMeasures.py DataArtificialSocialNetwork.txt***

- **Demo code:**

Trước tiên , tạo một mạng xã hội nhân tạo (Hình 4.1) phục vụ cho quá trình phân tích:

-Khai báo các thư viện:

```
import matplotlib.pyplot as plt
import networkx as nx
import sys
```

-Đọc dữ liệu đầu vào là file ***DataArtificialSocialNetwork.txt*** từ thư mục ***Resource***

```
G=nx.Graph()
with open("../Resource/"+sys.argv[1]) as f:
    lines = f.readlines()
    x = [int(line.split()[0]) for line in lines]
    y = [int(line.split()[1]) for line in lines]
    for i in range(len(x)):
        G.add_edge(x[i],y[i])
```

Với x[i] và y[i] là các nút đọc được từ file và tạo đồ thị G thông qua hàm add\_edge().



-Tạo một hàm để kiểm tra thông tin về số nút và cạnh của đồ thị:

```
def InformationOfGraph():
    print("InformationOfGraph:")
    print("Nodes: ",G.nodes())
    print("Edges: ",G.edges())
    print("Number of Nodes: ",G.number_of_nodes())
    print("Number of Edges: ",G.number_of_edges())
```

### Kết quả:

Nodes: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]

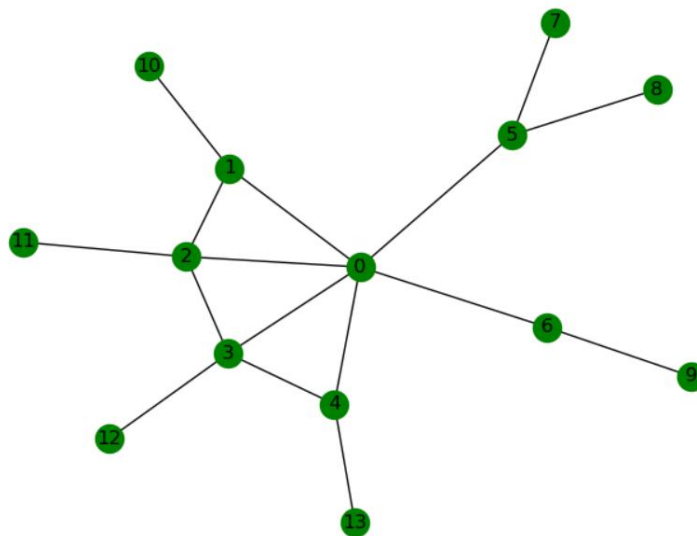
Edges: [(0, 1), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), (1, 2), (1, 10), (2, 3), (2, 11), (3, 4), (3, 12), (4, 13), (5, 7), (5, 8), (6, 9)]

Number of Nodes: 14

Number of Edges: 16

-Sau đó vẽ đồ thị G:

```
nx.draw(G, node_color='green', with_labels = True)
plt.show()
```



**Hình 4.1 Hình ảnh demo một mạng xã hội nhân tạo với 14 nút 16 cạnh**

#### 4.1.1 Demo về Mật độ (Density)

*def Density():*

```

    n=G.number_of_nodes()
    m=G.number_of_edges()
    D=m/(n*(n-1)/2)
    print("-----")
    print("Density of this network : ",D)
    print("-----")

```

-Với n là tổng số nút trong mạng (14) và m là tổng số cạnh trong mạng (16). Ta dễ dàng triển khai tính toán mật độ  $D=m/(n*(n-1)/2)$  của mạng xã hội này.

**Kết quả:**

```

-----
Density of this network :  0.17582417582417584
-----

```

#### 4.1.2 Demo về trung tâm độ (Degree centrality)

*def DegreeCentrality():*

```

    TempCd=[]
    Sum=0;
    for i in range(0,len(G.nodes())):
        m=len(G.edges(i))
        n=G.number_of_nodes()
        Cd=m/(n-1)
        Sum+=Cd
        TempCd.append(Cd);
    if(i<10):
        print("DegreeCentrality of Node",i," : ",Cd)

```

```

else:
    print("DegreeCentrality of Node",i," : ",Cd)
Avg=Sum/len(G.nodes())
Max=max(TempCd)
Min=min(TempCd)
a=[]
b=[]
for i in range(0,len(TempCd)):
    if(TempCd[i]==Max):
        a.append(i);
    if(TempCd[i]==Min):
        b.append(i);

print("-----")
print("Average DegreeCentrality : ",Avg)
print("Lowest DegreeCentrality : ",Min,"( Node",b,")")
print("Highest DegreeCentrality : ",Max,"( Node",a,")")
print("-----")

```

- Khai báo mảng TempCd=[] và biến Sum=0 , thông qua vòng lặp i chạy từ 0 đến độ dài của mảng chứa các nút trong mạng.
- Với m là tổng số các mối quan hệ trực tiếp của nút thứ i, n là tổng số các nút trong mạng xã hội. Tiến hành tính trung tâm độ của nút thứ i bằng biến  $Cd = m / (n - 1)$ .
- Sau đó chèn kết quả vào mảng TempCd.
- Biến Sum lúc này sẽ cộng tất cả giá trị của Cd để phục vụ cho việc tính toán giá trị trung tâm độ trung bình  $Avg = Sum / len(G.nodes())$ .
- Biến Max để tìm giá trị trung tâm độ lớn nhất và Min để tìm giá trị trung tâm độ nhỏ nhất trong mảng TempCd.

-Duyệt mảng TempCd , nếu phần tử thứ i trong mảng TempCd bằng giá trị Max hoặc Min thì gán thứ tự thứ i đó vào các biến a,b tương ứng. Khi đó , biến a,b chính là nút được xác định ứng với giá trị trung tâm độ lớn nhất và nhỏ nhất.

### **Kết quả:**

DegreeCentrality of Node 0 : 0.46153846153846156

DegreeCentrality of Node 1 : 0.23076923076923078

DegreeCentrality of Node 2 : 0.3076923076923077

DegreeCentrality of Node 3 : 0.3076923076923077

DegreeCentrality of Node 4 : 0.23076923076923078

DegreeCentrality of Node 5 : 0.23076923076923078

DegreeCentrality of Node 6 : 0.15384615384615385

DegreeCentrality of Node 7 : 0.07692307692307693

DegreeCentrality of Node 8 : 0.07692307692307693

DegreeCentrality of Node 9 : 0.07692307692307693

DegreeCentrality of Node 10 : 0.07692307692307693

DegreeCentrality of Node 11 : 0.07692307692307693

DegreeCentrality of Node 12 : 0.07692307692307693

DegreeCentrality of Node 13 : 0.07692307692307693

-----

Average DegreeCentrality : 0.17582417582417592

Lowest DegreeCentrality : 0.07692307692307693 ( Node [7, 8, 9, 10, 11, 12, 13] )

Highest DegreeCentrality : 0.46153846153846156 ( Node [0] )

-----

### 4.1.3 Demo về Trung tâm lân cận (Closeness centrality)

*def ClosenessCentrality():*

*closeness=nx.closeness\_centrality(G)*

*Sum=0;*

*TempCc=[]*

*for i in range(0,len(closeness)):*

*Sum+=closeness[i];*

*TempCc.append(closeness[i]);*

*if(i<10):*

*print("ClosenessCentrality of Node",i," : ",closeness[i])*

*else:*

*print("ClosenessCentrality of Node",i," : ",closeness[i])*

*Avg=Sum/len(G.nodes())*

*Max=max(TempCc)*

*Min=min(TempCc)*

*a=[]*

*b=[]*

*for i in range(0,len(TempCc)):*

*if(TempCc[i]==Max):*

*a.append(i);*

*if(TempCc[i]==Min):*

*b.append(i);*

*print("-----")*

*print("Average ClosenessCentrality : ",Avg)*

*print(" Lowest ClosenessCentrality : ",Min,"( Node",b,")")*

*print(" Highest ClosenessCentrality : ",Max,"( Node",a,")")*

*print("-----")*

Các công thức tương tự đều có thể thực hiện được bằng thủ công. Để rút ngắn quá trình thực hiện, một số hàm có sẵn của thư viện networkx sẽ được giới thiệu để triển khai.

-Khai báo mảng TempCc=[] và biến Sum=0, Biến closeness chứa các giá trị trung tâm lân cận thông qua hàm nx.closeness\_centrality(G) tích hợp sẵn trong thư viện.

-Vòng lặp i chạy từ 0 đến độ dài biến closeness. Sau đó chèn kết quả closeness[i] vào mảng TempCc. Biến Sum lúc này sẽ cộng tất cả giá trị của closeness[i] để phục vụ cho việc tính toán giá trị trung tâm lân cận trung bình  $Avg = \text{Sum} / \text{len}(G.\text{nodes}())$ .

-Biến Max để tìm giá trị trung tâm lân cận lớn nhất và Min để tìm giá trị trung tâm lân cận nhỏ nhất trong mảng TempCc.

-Duyệt mảng TempCc, nếu phần tử thứ i trong mảng TempCc bằng giá trị Max hoặc Min thì gán thứ tự thứ i đó vào các biến a,b tương ứng. Khi đó, biến a,b chính là nút được xác định ứng với giá trị trung tâm lân cận lớn nhất và nhỏ nhất.

### **Kết quả:**

ClosenessCentrality of Node 0 : 0.65

ClosenessCentrality of Node 1 : 0.4642857142857143

ClosenessCentrality of Node 2 : 0.5

ClosenessCentrality of Node 3 : 0.5

ClosenessCentrality of Node 4 : 0.4642857142857143

ClosenessCentrality of Node 5 : 0.4642857142857143

ClosenessCentrality of Node 6 : 0.43333333333333335

ClosenessCentrality of Node 7 : 0.325

ClosenessCentrality of Node 8 : 0.325

ClosenessCentrality of Node 9 : 0.30952380952380953

ClosenessCentrality of Node 10 : 0.325

ClosenessCentrality of Node 11 : 0.34210526315789475

ClosenessCentrality of Node 12 : 0.34210526315789475

ClosenessCentrality of Node 13 : 0.325

-----

Average ClosenessCentrality : 0.41213748657357685

Lowest ClosenessCentrality : 0.30952380952380953 ( Node [9] )

Highest ClosenessCentrality : 0.65 ( Node [0] )

-----

#### **4.1.4 Demo về Trung tâm trung gian (Betweenness centrality)**

*def BetweennessCentrality():*

*betweenness=nx.betweenness\_centrality(G)*

*Sum=0;*

*TempCb=[]*

*for i in range(0,len(betweenness)):*

*Sum+=betweenness[i];*

*TempCb.append(betweenness[i]);*

*if(i<10):*

*print("BetweennessCentrality of Node",i," : ",betweenness[i])*

*else:*

*print("BetweennessCentrality of Node",i," : ",betweenness[i])*

*Avg=Sum/len(G.nodes())*

*Max=max(TempCb)*

*Min=min(TempCb)*

*a=[]*

*b=[]*

*for i in range(0,len(TempCb)):*

*if(TempCb[i]==Max):*

```

        a.append(i);
        if(TempCb[i]==Min):
            b.append(i);

    print("-----")
    print("Average BetweennessCentrality : ",Avg)
    print("Lowest BetweennessCentrality : ",Min,"( Node",b,")")
    print("Highest BetweennessCentrality : ",Max,"( Node",a,")")
    print("-----")

```

-Khai báo mảng TempCb=[] và biến Sum=0 , Biến betweenness chứa các giá trị trung tâm trung gian thông qua hàm nx.betweenness\_centrality(G) tích hợp sẵn trong thư viện.

-Vòng lặp i chạy từ 0 đến độ dài biến betweenness . Sau đó chèn kết quả closeness[i] vào mảng TempCb. Biến Sum lúc này sẽ cộng tất cả giá trị của closeness[i] để phục vụ cho việc tính toán giá trị trung tâm trung gian trung bình Avg=Sum/len(G.nodes()).

-Biến Max để tìm giá trị trung tâm trung gian lớn nhất và Min để tìm giá trị trung tâm trung gian nhỏ nhất trong mảng TempCb.

-Duyệt mảng TempCb , nếu phần tử thứ i trong mảng TempCb bằng giá trị Max hoặc Min thì gán thứ tự thứ i đó vào các biến a,b tương ứng. Khi đó , biến a,b chính là nút được xác định ứng với giá trị trung tâm trung gian lớn nhất và nhỏ nhất.

### **Kết quả:**

BetweennessCentrality of Node 0 : 0.6923076923076923

BetweennessCentrality of Node 1 : 0.15384615384615385

BetweennessCentrality of Node 2 : 0.1794871794871795

BetweennessCentrality of Node 3 : 0.1794871794871795

BetweennessCentrality of Node 4 : 0.15384615384615385



BetweennessCentrality of Node 5 : 0.2948717948717949  
 BetweennessCentrality of Node 6 : 0.15384615384615385  
 BetweennessCentrality of Node 7 : 0.0  
 BetweennessCentrality of Node 8 : 0.0  
 BetweennessCentrality of Node 9 : 0.0  
 BetweennessCentrality of Node 10 : 0.0  
 BetweennessCentrality of Node 11 : 0.0  
 BetweennessCentrality of Node 12 : 0.0  
 BetweennessCentrality of Node 13 : 0.0

-----  
 Average BetweennessCentrality : 0.12912087912087913  
 Lowest BetweennessCentrality : 0.0 ( Node [7, 8, 9, 10, 11, 12, 13] )  
 Highest BetweennessCentrality : 0.6923076923076923 ( Node [0] )  
 -----

#### ***4.1.5 Demo về Trung tâm phân cụm (Clustering centrality)***

*def ClusteringCentrality():*

*clustering=nx.clustering(G)*

*Sum=0;*

*TempCclustering=[]*

*for i in range(0,len(clustering)):*

*Sum+=clustering[i];*

*TempCclustering.append(clustering[i]);*

*if(i<10):*

*print("ClusteringCentrality of Node",i," : ",clustering[i])*

*else:*

*print("ClusteringCentrality of Node",i," : ",clustering[i])*

```

Avg=Sum/len(G.nodes())
Max=max(TempCclustering)
Min=min(TempCclustering)
a=[]
b=[]
for i in range(0,len(TempCclustering)):
    if(TempCclustering[i]==Max):
        a.append(i);
    if(TempCclustering[i]==Min):
        b.append(i);

print("-----")
print("Average ClusteringCentrality : ",Avg)
print("Lowest ClusteringCentrality : ",Min,"( Node",b,")")
print("Highest ClusteringCentrality : ",Max,"( Node",a,")")
print("-----")

```

(Trương tự như các ví dụ đã nêu).

### **Kết quả:**

```

ClusteringCentrality of Node 0 : 0.2
ClusteringCentrality of Node 1 : 0.3333333333333333
ClusteringCentrality of Node 2 : 0.3333333333333333
ClusteringCentrality of Node 3 : 0.3333333333333333
ClusteringCentrality of Node 4 : 0.3333333333333333
ClusteringCentrality of Node 5 : 0
ClusteringCentrality of Node 6 : 0
ClusteringCentrality of Node 7 : 0
ClusteringCentrality of Node 8 : 0

```

ClusteringCentrality of Node 9 : 0

ClusteringCentrality of Node 10 : 0

ClusteringCentrality of Node 11 : 0

ClusteringCentrality of Node 12 : 0

ClusteringCentrality of Node 13 : 0

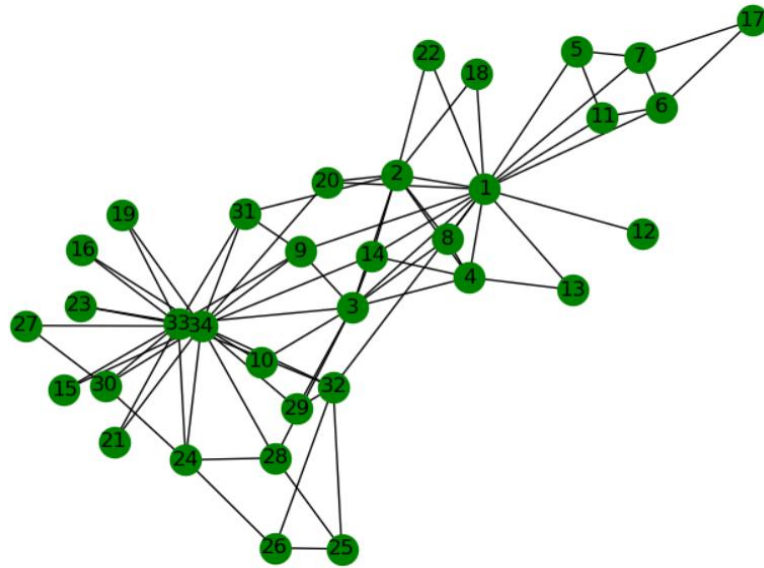
-----  
Average ClusteringCentrality : 0.10952380952380951

Lowest ClusteringCentrality : 0 ( Node [5, 6, 7, 8, 9, 10, 11, 12, 13] )

Highest ClusteringCentrality : 0.3333333333333333 ( Node [1, 2, 3, 4] )  
-----

Ở bài báo cáo này chỉ liệt kê một vài thông số chính , các hệ số , cao nhất, thấp nhất, trung bình của mạng. Có thể dựa vào một vài thông số ở trên , người chủ chốt (key player) lúc này thuộc về nút 0. Do đó, nút 0 trong mạng xã hội này có tầm ảnh hưởng , có tác động to lớn đến toàn bộ mạng xã hội. Tuy nhiên, các số liệu liên quan bằng nhiều hình thức thống kê , đo lường khác nhau đều tùy thuộc vào các nhà phân tích mạng xã hội.

## 4.2 Demo về một thuật toán nhận diện cộng đồng

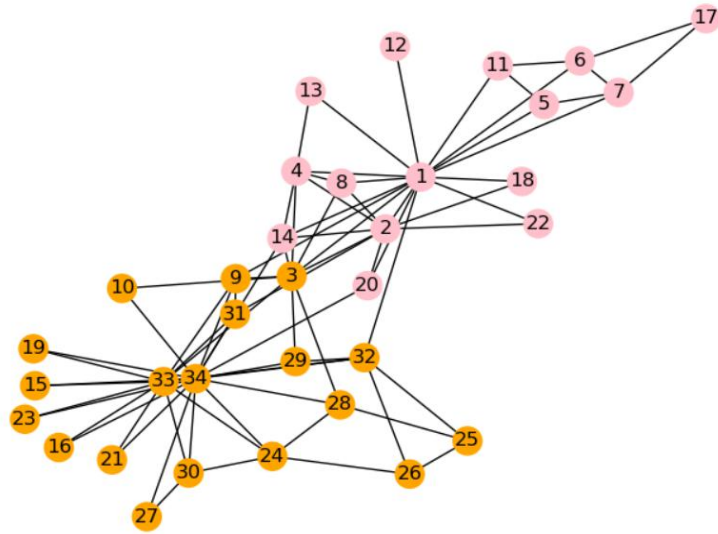


**Hình 4.2 Hình ảnh mô phỏng cho việc demo thuật toán Girvan Newman**

(Nguồn: Câu lạc bộ karate Zachary - Zachary W. 1977)

Đây là một đồ thị về câu lạc bộ Karate của Zachary được sử dụng bởi Michelle Girvan và Mark Newman trong việc phân tích nhận diện cộng đồng (2002). Mạng này gồm 34 thành viên, ghi lại mối liên kết giữa các cặp thành viên tương tác bên ngoài câu lạc bộ. Trong quá trình nghiên cứu, một cuộc xung đột nảy sinh giữa quản trị viên (nút 34) và người hướng dẫn (nút 1), dẫn đến việc chia câu lạc bộ thành hai. Việc mô phỏng quá trình phân tách câu lạc bộ này được triển khai bằng thuật toán của Girvan-Newman trong việc nhận diện cộng đồng.

Trước khi đi vào chi tiết cách thuật toán này hoạt động. Hãy xem kết quả thu được khi thực nghiệm phân tách và nhận diện cộng đồng bằng thuật toán Girvan-Newman của đồ thị này như sau:



**Hình 4.3** Kết quả phân cụm nhận diện cộng đồng (Câu lạc bộ karate Zachary)

Sử dụng lại mạng xã hội nhân tạo (file *DataArtificialSocialNetwork.txt*) làm tập dữ liệu đầu vào để demo cho thuật toán này.

#### Demo code:

-Khai báo các thư viện:

```
import matplotlib.pyplot as plt
import networkx as nx
import sys
```

-Đọc dữ liệu đầu vào là file *DataArtificialSocialNetwork.txt* từ thư mục *Resource*

```
G=nx.Graph()
with open("../Resource/"+sys.argv[1]) as f:
    lines = f.readlines()
    x = [int(line.split()[0]) for line in lines]
    y = [int(line.split()[1]) for line in lines]
    for i in range(len(x)):
        G.add_edge(x[i],y[i])
```

- Với  $x[i]$  và  $y[i]$  là các nút đọc được từ file và tạo đồ thị  $G$  thông qua hàm `add_edge()`.
- Tìm cạnh có tính trung tâm trung gian cao và dùng thuật toán Girvan-Newman để loại bỏ cạnh đó, phân tách và nhận diện cộng đồng qua 2 hàm sau:

```
def HighCb(G):
    Eb = nx.edge_betweenness centrality(G);
    E = ();
    for u, v in sorted(Eb.items(), key=lambda i: i[1], reverse = True):
        E = u;
        break;
    return E;

def GNA(G):
    links = nx.connected_components(G);
    number = nx.number_connected_components(G);
    while(number == 1):
        G.remove_edge(HighCb(G)[0], HighCb(G)[1]);
        links = nx.connected_components(G);
        number = nx.number_connected_components(G);
    return links;
```

- Tại hàm `HighCb(G)`, tìm hệ số trung tâm trung gian của các cạnh thông qua hàm `nx.edge_betweenness centrality(G)` cho vào biến `Eb`.
- Dùng hàm `sorted()` để sắp xếp thứ tự sau đó xuất danh sách cạnh.
- Tại hàm `GNA(G)`, Khai báo biến `links` dùng để tạo liên kết cho bộ các nút bằng `nx.connected_components`. Biến `number` để đếm số.
- Triển khai thuật toán của Girvan-Newman, bằng cách so sánh các cạnh có tính trung tâm trung gian lớn nhất và loại bỏ chúng - hàm `remove_edge()`.
- Xuất bộ liên kết chính là các phân chia vùng.

```

def Community():
    Temp = GNA(G.copy());
    Area = [];
    for i in Temp:
        Area.append(list(i));
    return Area;

```

-Sau khi phân chia vùng đã hoàn thành , tiến hành thao tác chèn các nút vào trong một mảng mới thông qua biến Area.

```

def ShowGraphAfterUseGNA():
    color = [];
    for i in G:
        if i in Community()[0]:
            color.append('pink');
        else:
            color.append('orange');
    nx.draw(G, node_color=color, with_labels=True);
    plt.show();
def ShowGraphAtFirst():
    nx.draw(G, node_color='green', with_labels = True);
    plt.show();

```

-Hàm ShowGraphAtFirst() in biểu đồ ban đầu bằng các nút màu xanh lá. Hàm ShowGraphAfterUseGNA(), xét từng phần tử trong biểu đồ G ban đầu , nếu gặp phần tử thứ i trong đó trùng với mảng Area trong hàm Community() , thì tô màu hồng , ngược lại tô màu cam. Việc còn lại lúc này là xem biểu đồ kết quả.

```

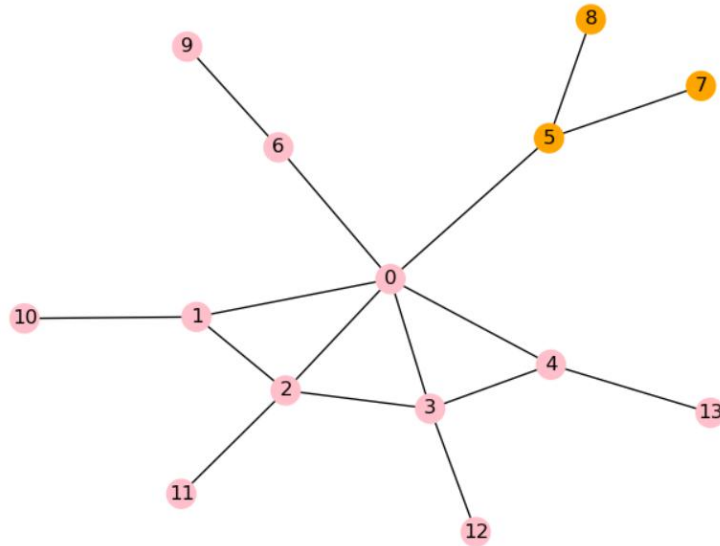
ShowGraphAfterUseGNA();

```

-Cuối cùng, thực thi câu lệnh trong terminal tại thư mục **SourceCode**.

***python 51800408\_CommunityDetection.py DataArtificialSocialNetwork.txt***

Ta được kết quả của quá trình phân cụm bằng thuật toán Girvan-Newman như Hình 4.4



**Hình 4.4 Kết quả phân cụm nhận diện cộng đồng (Mạng nhân tạo hình 4.1)**

Có thể so sánh những phần tô màu giữa Hình 4.3 và Hình 4.4. Thuật toán Girvan-Newman hoàn toàn dựa vào thông số có được từ các cạnh có tính trung gian và tìm ra cạnh trung gian lớn nhất. Đối với mạng xã hội nhân tạo (Hình 4.1), cạnh có số lần được di chuyển qua (làm trung gian) nhiều nhất chính là cạnh (0-5). Nếu ta cắt đứt cạnh trung gian này sẽ tạo thành 2 cụm (hay còn gọi là 2 mạng xã hội riêng biệt). Tương ứng với đồ thị Hình 4.1, ta có thể thay đổi thành vô vàn các tập dữ liệu khác nữa (Giả sử như tập dữ liệu câu lạc bộ Zachary - Hình 4.2) để phục vụ cho việc phân tích nhận diện cộng đồng.



Ý nghĩa mà thuật toán do Michelle Girvan và Mark Newman đem lại là rất to lớn cho các nhà phân tích trong việc nhận diện cộng đồng , giúp cho việc xác định người chủ chốt , người có tầm ảnh hưởng đến mạng. Khi một mối quan hệ bị cắt đứt , có thể dẫn đến một mạng xã hội bị tan rã.

## CHƯƠNG 5 – TỔNG KẾT

Bài báo cáo này đã nêu ra các khái niệm, ý nghĩa, công thức, thuật toán và những vấn đề trong mạng lưới xã hội. Khi phân tích bất kì mạng xã hội nào ta cũng cần triển khai phân tích theo nhiều hướng , nhiều phương pháp khác nhau. Điều đầu tiên cần làm chính là xác định mục đích của việc phân tích (ta đang phân tích về vấn đề gì , nhu cầu gì,..). Tiếp theo, dùng các thuật toán nhận diện cộng đồng , dùng các hệ số để đo lường mạng lưới xã hội đó , so sánh các yếu tố,... Sau đó tìm ra những người chủ chốt (key players), các tác động tích cực hay tiêu cực khi mất xích này bị thay đổi đột ngột hoặc theo một thời gian nhất định nào đó và rất nhiều vấn đề khác liên quan.

Việc ứng dụng phân tích mạng xã hội vào đời sống thực tế giúp ích rất nhiều cho quá trình hình thành và phát triển của một xã hội. Định hướng một tương lai mới , khả năng kiểm soát những biến đổi trong tương lai sẽ luôn là vấn đề nóng và hàng đầu cho các nhà phân tích mạng xã hội.

## TÀI LIỆU THAM KHẢO

- [1] Scott, J. (1988). Social Network Analysis. *Sociology*, 22(1), 109–127.
- [2] Steketee, M., Miyaoka, A., & Spiegelman, M. (2015). Social Network Analysis. *International Encyclopedia of the Social & Behavioral Sciences*, 461–467.
- [3] Otte, E., & Rousseau, R. (2002). Social network analysis: a powerful strategy, also for the information sciences. *Journal of Information Science*, 28(6), 441–453.
- [4] Berzinji, A., Kaati, L., & Rezine, A. (2012). Detecting Key Players in Terrorist Networks. 2012 European Intelligence and Security Informatics Conference.
- [5] Girvan, M., & Newman, M.E.J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12), 7821–7826
- [6] Wasserman, S., & Faust, K. (n.d.). Dyads. *Social Network Analysis*, 505–555.
- [7] Raghavan, U. N., Albert, R., & Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3).
- [8] Leicht, E. A., & Newman, M. E. J. (2008). Community Structure in Directed Networks. *Physical Review Letters*, 100(11).
- [9] Zachary W. (1977). An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33, 452-473.
- [10] Hamers, L., Hemeryck, Y., Herweyers, G., Janssen, M., Keters, H., Rousseau, R., & Vanhoutte, A. (1989). Similarity measures in scientometric research: The Jaccard index versus Salton's cosine formula. *Information Processing & Management*, 25(3), 315–318.
- [11] Goldberg, D.S., & Roth, F. P. (2003). Assessing experimentally derived interactions in a small world. *Proceedings of the National Academy of Sciences*, 100(8), 4372–4376.
- [12] Aric Hagberg, Dan Schult, Pieter Swart (2020). *NetworkX Reference*. 195-241.

## TỰ ĐÁNH GIÁ

**BẢNG TỰ ĐÁNH GIÁ**

Yêu cầu	Điểm /10	Cấp độ 1	Cấp độ 2	Cấp độ 3	Tự đánh giá	Lý do
		0 điểm	1/2 điểm	Trọn điểm		
<b>1/ Báo cáo</b>	<b>8.0</b>					
Đúng định dạng	1.0	Sai định dạng và phác thảo	Có một vài lỗi	Đúng định dạng và phác thảo, không lỗi	1.0	
Chương 1	1.0	Không đủ nội dung, viết xấu, không có ví dụ	Nội dung đầy đủ, không được viết tốt, không đủ ví dụ	Nội dung đầy đủ, được viết tốt với các ví dụ	1.0	
Chương 2	2.0	Không đủ nội dung, viết xấu, không có ví dụ	Nội dung đầy đủ, không được viết tốt, không đủ ví dụ	Nội dung đầy đủ, được viết tốt với các ví dụ	2.0	
Chương 3	2.0	Không đủ nội dung, viết xấu, không có ví dụ	Nội dung đầy đủ, không được viết tốt, không đủ ví dụ	Nội dung đầy đủ, được viết tốt với các ví dụ	2.0	
Chương 4	1.0	Không đủ nội dung, viết xấu, không có ví dụ	Nội dung đầy đủ, không được viết tốt, không đủ ví dụ	Nội dung đầy đủ, được viết tốt với các ví dụ	1.0	
Chương 5	0.5	Không đủ nội dung, viết xấu, không có ví dụ	Nội dung đầy đủ, không được viết tốt	Nội dung đầy đủ, được viết tốt	0.25	

<b>Yêu cầu</b>	<b>Điểm /10</b>	<b>Cấp độ 1</b>	<b>Cấp độ 2</b>	<b>Cấp độ 3</b>	<b>Tự đánh giá</b>	<b>Lý do</b>
Tài liệu tham khảo	0.5	Không có tài liệu tham khảo	Sai định dạng, < 3 tài liệu tham khảo	Đúng định dạng, $\geq 3$ tài liệu tham khảo	0.5	
<b>2/ Demo</b>	<b>2.0</b>					
Nội dung	1.0	Demo $\leq 1$ công thức /thuật toán	Demo được 3 công thức hoặc thuật toán	Demo $\geq 5$ công thức hoặc thuật toán	1.0	
Chương trình	1.0	Không thể biên dịch	Lỗi thời gian chạy	Chạy chính xác và không lỗi	1.0	
<b>Tổng điểm</b>	10.0	Kết quả:			9.75	