**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH   
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

***PHẠM NGUYỄN MỸ DIỄM  
PHẠM NGỌC TÂN***

**TECHNICAL REPORT**

**LAB 01:**

**XÂY DỰNG CÔNG CỤ CHO PHÉP BIỂU DIỄN BIG GRAPH VÀ HÀM TRÍCH ĐỒ THỊ CON**

**PHƯƠNG PHÁP TOÁN CHO TRÍ TUỆ NHÂN TẠO – CAO HỌC K32**

**Giảng viên lý thuyết:**

**PGS.TS NGUYỄN ĐÌNH THÚC**

**TP. Hồ Chí Minh, năm 2023**

**Project Team**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Mã số học viên | Họ tên | Email liên hệ |
| 1 | 22C11029 | Phạm Nguyễn Mỹ Diễm | [22c11029@student.hcmus.edu.vn](mailto:22c11029@student.hcmus.edu.vn) |
| 2 | 22C15042 | Phạm Ngọc Tân | [22c15042@student.hcmus.edu.vn](mailto:22c15042@student.hcmus.edu.vn) |

**Nội dung**

[Giới thiệu 2](#_Toc388620531)

[Tổng quan 3](#_Toc388620532)

[Phương thức 4](#_Toc388620533)

[DirectedGraph: Đồ thi có hướng 4](#_Toc388620534)

[UndirectedGraph: Đồ thi vô hướng 5](#_Toc388620534)

[Chạy thử trên một số testing 5](#_Toc388620552)

[Source code 7](#_Toc388620552)

[Tham khảo 7](#_Toc388620553)

# Giới thiệu

Hiện nay, có rất nhiều công cụ để biểu diễn đồ thị lớn với rất nhiều loại dữ liệu được xây dựng từ những công cụ này. Với sự tiến bộ đó, nhóm đã xây dựng công cụ để biểu diễn đồ thị lớn tên BigGraphDT, một lập trình ứng dụng (API) cung cấp thư viện để dựng và quản lý đồ thị lớn.

Nhóm đã sử dụng dữ liệu từ Stanford Large Network Dataset Collection để thử nghiệm trên API mà nhóm đã cung cấp. Cụ thể, dữ liệu wikipedia vote network nhóm đang sử dung này được lấy từ trang Wikipedia (từ ngày 3 tháng 1 năm 2008). Dữ liệu wiki vote từ khi thành lập Wikipedia cho đến tháng 1 năm 2008. Các đỉnh trong mạng thể hiện cho người dùng wikipedia và cạnh có hướng từ node i đến node j thể hiện cho người i đã bình chọn hay bỏ phiếu cho người j.

Ngoài ra, ta cũng có thể linh hoạt sử dụng nhiều dataset khác từ SNAP: Network datasets và nhiều loại dataset từ những nguồn khác.

# Tổng quan

class **DirectedGraph**(copy\_graph: nx.DiGraph() = None, graph\_data\_file: str = None)

* Lớp cơ sở cho đồ thị có hướng.
* **DirectedGraph** lưu trữ các đỉnh và cạnh với dữ liệu và thuộc tính tùy chọn.
* Các cạnh được biểu diễn dưới dạng liên kết giữa các đỉnh với các thuộc tính key/value tùy chọn.

class **UndirectedGraph**(DirectedGraph)

Lớp cơ sở cho đồ thị vô hướng.

Ví dụ: Ta đọc file input cho đồ thị có hướng như bên dưới (hình 1), là các danh sách kề bao gồm các đinh và các đỉnh kề tương ứng.

G = **Graph**(graph\_data\_file='sample\_biggraph\_data.txt', is\_directed=True)

Text

Description automatically generated

**Hình 1.** Danh sách kề bao gồm các đỉnh và các đỉnh kề tương ứng từ dataset wiki vote.

Trực quan hóa đồ thị lớn từ dataset wiki vote (hình 2):

G.**visualization**(save\_path="visualization/big\_graph.png",verbose=0,figsize\_h=12, figsize\_w=10)

Chart, bubble chart

Description automatically generatedChart, radar chart

Description automatically generated

**Hình 2.** Biểu diễn đồ thị lớn với dữ liệu wiki vote. **Hình 3**. Biểu diễn đồ thị con gồm một số node từ đồ thị lớn (hình 2) của dữ liệu wiki vote.

Bây giờ, ta sẽ trích đồ thị con bằng cách gọi hàm **get\_subgraph** và trực quan hóa kết quả (hình 3) như sau:

sub\_graph=**Graph**(copy\_graph=G.**get\_subgraph**(['3', '28', '182', '214', '271', '286', '300', '348', '349', '371', '567', '581', '584', '590', '604', '611', '8283']))

sub\_graph.**visualization**(scale\_coefficient=3.5, save\_path='visualization/sub\_biggraph.png', verbose=0, figsize\_h=16, figsize\_w=10)

print(sub\_graph)

Một số thuộc tính của đồ thị được liệt kê bên dưới như: số đỉnh, cạnh, liệt kê các đỉnh, cạnh, bậc của đỉnh tương ứng của đồ thị (hình 4).

Text

Description automatically generated

Hình 4. Một số thuộc tính của đồ thị con được rút trích ở trên gồm: số đỉnh, cạnh, liệt kê các đỉnh, cạnh, bậc của đỉnh tương ứng của đồ thị.

# Phương thức

Có 2 loại đồ thị được định nghĩa bên dưới:

## DirectedGraph: Đồ thi có hướng

Thêm và xóa các đỉnh, cạnh

|  |  |
| --- | --- |
| DirectedGraph**.\_\_init\_\_**(copy\_graph:DiGraph(), graph\_data\_file:str) | Khởi tạo đồ thị với các thuộc tính đỉnh, cạnh của đồ thị và file input của đồ thị tương ứng |
| DirectedGraph.**addEdge**(source\_node:str, \*linked\_node) | Hàm thêm các đỉnh kề vào đỉnh hiện tại |
| DirectedGraph.**removeNode**(node\_to\_remove:str) | Hàm xóa một đỉnh từ đồ thị |
| DirectedGraph.**removeEdges**(u:str, v:str) | Hàm xóa một cạnh từ đồ thị |

Xuất thông tin các đỉnh và cạnh

|  |  |
| --- | --- |
| DirectedGraph.**get\_adjacency\_list(**) | Hàm lấy đồ thị dưới dạng danh sách kề |
| DirectedGraph.**printNodes**() | Hàm in ra tất cả các đỉnh từ đồ thị |
| DirectedGraph.**printEdges**() | Hàm in ra tất cả các cạnh từ đồ thị |

Đếm bậc các đỉnh

|  |  |
| --- | --- |
| DirectedGraph.**getNodeDegree**(node:str) | Hàm lấy bậc của một đỉnh |

Trích đồ thị con

|  |  |
| --- | --- |
| DirectedGraph.**get\_subgraph**(nodes\_to\_retrieve:List[str]) | Hàm trích đồ thị con từ đồ thị lớn bằng cách truy xuất một đỉnh |

Trực quan hóa đồ thị

|  |  |
| --- | --- |
| DirectedGraph.**visualization**(figsize\_h: float, figsize\_w:float, scale\_coefficient:float, save\_path:str, verbose:bool) | Hàm trực quan hóa đồ thị với một số các tham số |

## UndirectedGraph: Đồ thị vô hướng

Kế thừa các phương phức của đồ thị có hướng được cung cấp ở trên.

# Chạy thử trên một số testing

Testing 1:

G = **Graph**(graph\_data\_file='example\_data1.txt', is\_directed=True)

print(f"[Result] Properties of graph")

print(G)

G.**visualization**(scale\_coefficient=3.5,save\_path="visualization/example1\_graph.png", verbose=0, figsize\_h=12, figsize\_w=10)

Chart

Description automatically generatedGraphical user interface, text

Description automatically generated

Hình 5a. Đồ thị biểu diễn đồ thị có hướng.

Hình 5b. Danh sách kề bao gồm các đỉnh và đỉnh kề tương ứng của đồ thị bên trái.

Output:

Text

Description automatically generatedc

Hình 5c. Thông tin một số thuộc tính của đồ thị được liệt kê như: số đỉnh, cạnh, liệt kê các đỉnh, cạnh, bậc của đỉnh tương ứng của đồ thị.

Testing 2:

G = **Graph**(graph\_data\_file='data/example\_data2.txt', is\_directed=False)

print(f"[Result] Properties of graph")

print(G)

G.**visualization**(scale\_coefficient=3.5, save\_path="visualization/example2\_graph.png", verbose=0, figsize\_h=12, figsize\_w=10)

Chart, line chart

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Hình 6a. Đồ thị biểu diễn đồ thị vô hướng.

Hình 6b. Danh sách kề bao gồm các đỉnh và đỉnh kề tương ứng của đồ thị bên trái.

Output:

Text

Description automatically generated

Hình 6c. Thông tin một số thuộc tính của đồ thị được liệt kê như: số đỉnh, ccạnh, liệt kê các đỉnh, cạnh, bậc của đỉnh tương ứng của đồ thị.

# Source code

Toàn bộ code và data có thể xem tại đường link:

# Tham khảo

[1] SNAP: Network datasets: Wikipedia vote network

[2] DiGraph—Directed graphs with self loops – NetworkX 3.0 documentation