**CHƯƠNG TRÌNH THỰC NGHIỆM BÀI TOÁN TỐI ƯU ẢNH HƯỞNG TRÊN MẠNG XÃ HỘI SỬ DỤNG THUẬT TOÁN CELF VÀ RIS**

**Giới thiệu**

Chương trình này được thiết kế để thực hiện bài toán tối đa hóa ảnh hưởng (Influence Maximization) trên một đồ thị mạng xã hội. Chương trình sử dụng hai thuật toán chính là:

* **CELF (Cost-Effective Lazy Forward)**
* **RIS (Reverse Influence Sampling)**

Chương trình sẽ tạo một đồ thị mạng xã hội, chạy các thuật toán tối đa hóa ảnh hưởng, và xuất kết quả dưới dạng văn bản và hình ảnh.

**Tải chương trình demo:**

Clone repository hoặc tải file nén từ địa chỉ

https://github.com/HieuHuynhDev30/Demo\_toi\_da\_anh\_huong\_MXH.git

**Yêu cầu hệ thống**

* Python 3.7 hoặc cao hơn.
* Các thư viện Python:
  + numpy
  + pandas
  + matplotlib
  + networkx
  + collections
  + time
  + pycairo

**Hướng dẫn sử dụng**

**1. Chạy chương trình**

* Sau khi clone repository hoặc tải về từ trang github của chương trình, truy cập vào thư mục chương trình chứa file demo.py và mở Command Prompt
* Kích hoạt môi trường ảo để cô lập chương trình khỏi cài đặt của hệ thống, đảm bảo chương trình thực thi ổn định hoặc có thể bỏ qua bước này:

cd .\venv\Scripts\

.\activate

* Cài đặt các thư viện cần thiết nếu chưa có:

pip install numpy pandas matplotlib networkx igraph pycairo

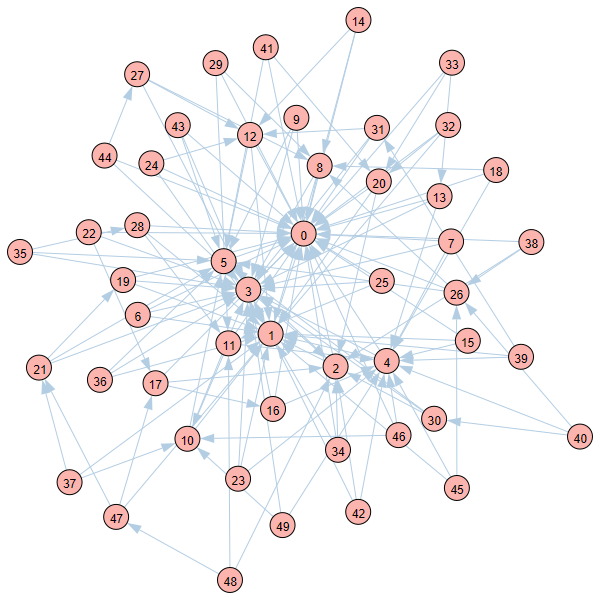
* Quay trở lại thư mục chứa file demo.py và thực thi chương trình

python demo.py

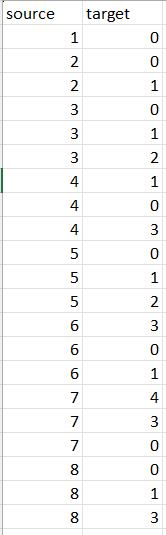
**2. Các bước thực hiện trong chương trình**

**Bước 1: Tạo đồ thị mạng xã hội**

* Chương trình yêu cầu nhập:
  + **Số nút**: Số lượng nút trong đồ thị.
  + **Số cạnh được kết nối từ mỗi nút mới**: Mỗi nút mới sẽ kết nối với bao nhiêu nút hiện có.
* Ví dụ nhập liệu:
  + Nhập số nút: 100
  + Nhập số cạnh được kết nối từ mỗi nút mới: 3
* Nếu nhập đúng, chương trình sẽ tạo đồ thị và lưu đồ thị vào file G\_figure.png, lưu danh sách các cạnh của đồ thị vào G\_dataframe.csv. Ví dụ:



Hình 1. Đồ thị G



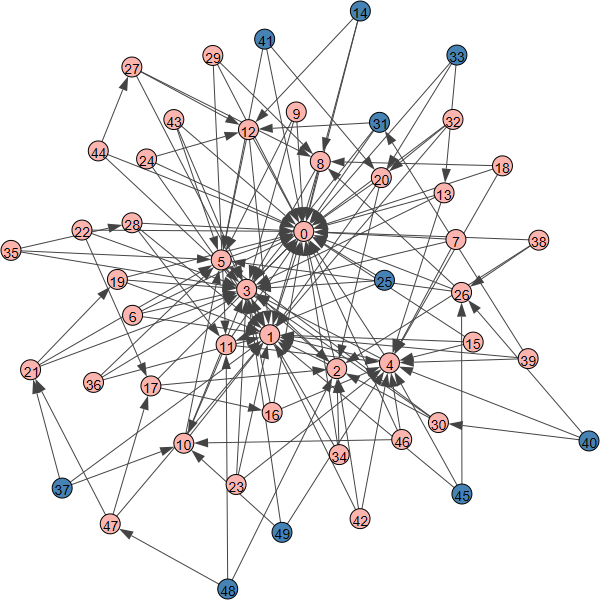
Hình 2. File csv lưu danh sách các cạnh của đồ thị

* Tắt cửa sổ hiển thị đồ thị G để tiếp tục chương trình

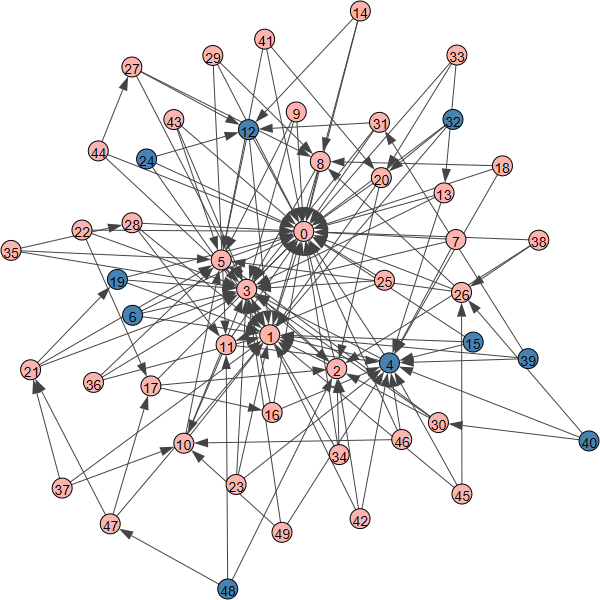
**Bước 2: Nhập thông số cho các thuật toán**

* Chương trình yêu cầu nhập:
  + **Số lượng đỉnh trong tập hạt giống (k)**: Số nút cần chọn để tối đa hóa ảnh hưởng.
  + **Xác suất lan truyền (p)**: Một giá trị giữa 0 và 1 đại diện cho xác suất lan truyền ảnh hưởng.
  + **Số lần mô phỏng Monte Carlo (mc)**: Số lần mô phỏng để ước tính ảnh hưởng.
* Ví dụ nhập liệu:
  + Số lượng đỉnh trong tập hạt giống: 5
  + Xác suất lan truyền p (0 < p < 1): 0.1
  + Số lần mô phỏng Monte Carlo: 1000

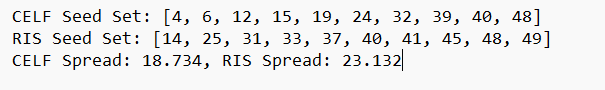
**Bước 3: Kết quả và xuất đồ thị**

* Sau khi chạy các thuật toán, chương trình sẽ hiển thị kết quả:
  + **Tập hạt giống của CELF và RIS**.
  + **Ảnh hưởng lan truyền của từng tập hạt giống**.
* Các kết quả sẽ được lưu vào các file:
  + CELF\_&\_RIS\_results.txt: Chứa tập hạt giống và ảnh hưởng lan truyền.
  + CELF\_graph.png: Đồ thị kết quả từ thuật toán CELF.
  + RIS\_graph.png: Đồ thị kết quả từ thuật toán RIS.
  + computation\_time\_chart.png: Biểu đồ thời gian tính toán của hai thuật toán.
* Ví dụ các file kết quả được lưu:

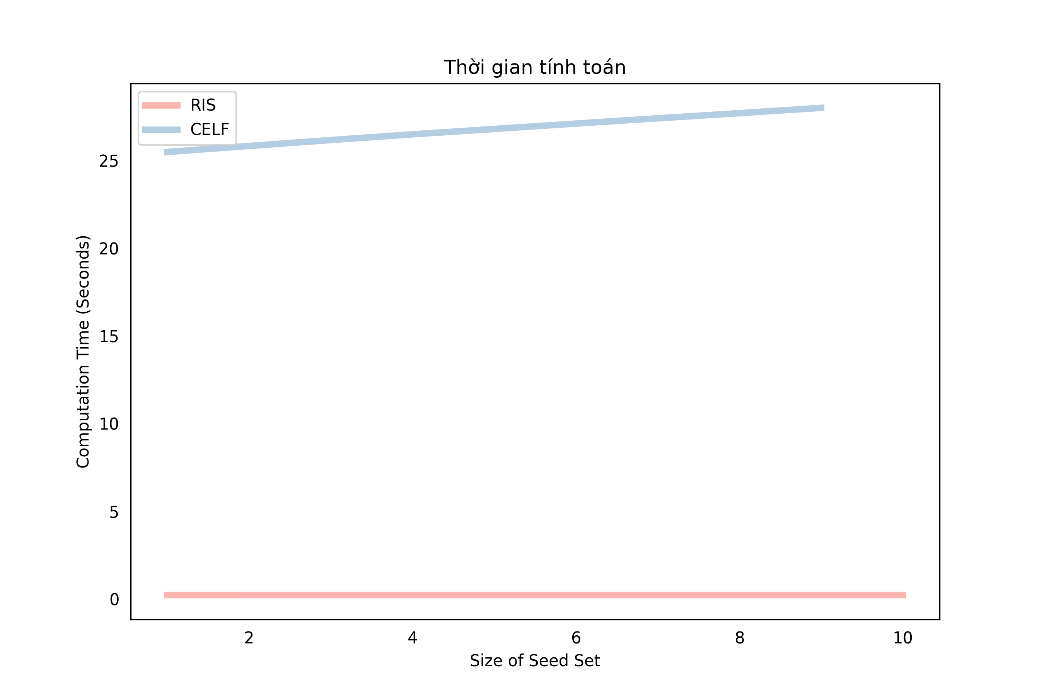
Hình 4. Đồ thị G với các nút hạt giống từ thuật toán RIS



Hình 3. Đồ thị G với các nút hạt giống từ thuật toán CELF



Hình 2. Tập hạt giống và kết quả hàm ảnh hưởng của các thuật toán



Hình 5. Biểu đồ so sánh thời gian tính toán của hai thuật toán

**Kết quả lưu trữ**

* **File văn bản**: Chứa kết quả chi tiết của các thuật toán.
* **Hình ảnh**:
  + Đồ thị mạng xã hội với các nút trong tập hạt giống được đánh dấu.
  + Biểu đồ so sánh thời gian tính toán giữa CELF và RIS.

**Ghi chú**

* Đảm bảo nhập đúng định dạng yêu cầu để tránh lỗi.
* Các kết quả được lưu trong thư mục hiện tại nơi chương trình được chạy.
* Nếu nhập sai định dạng của các thông số đầu vào, chương trình sẽ yêu cầu nhập lại cho đến khi đúng.
* Thoát chương trình khi chưa thực thi xong bằng tổ hợp Ctrl + C.