

Graph Theory

Bài tập lớn

Ngày 23 tháng 3 năm 2016

1 Yêu cầu bắt buộc về ngôn ngữ lập trình

Sinh viên chỉ được phép chọn một trong hai ngôn ngữ C++ hoặc Java và chỉ được phép sử dụng thư viện chuẩn. Tuyệt đối không được phép sử dụng thêm bất kỳ thư viện hoặc gói hỗ trợ khác.

2 Yêu cầu nộp bài và cấm thi

1. **Loại bài tập:** cá nhân

2. **Deadline:** 11/4/2016

3. **Nộp bài**

- (a) Sinh viên đăng nhập vào hệ thống Sakai (sakai.it.tdt.edu.vn) môn Graph Theory bằng tài khoản được cung cấp trước đó.
- (b) Thư mục *MSSV_HoTen_NX* được nén thành file .rar hoặc .zip. Với X tương ứng với Nhóm lý thuyết. Thư mục nộp bài gồm:
 - Source chứa tất cả các file liên quan đến nội dung bài tập lớn
 - Release chứa file *MSSV_HoTen_NX.exe* hoặc *MSSV_HoTen_NX.jar* để thực thi chương trình
 - Readme.doc: thông tin cách build và thực thi, hướng dẫn sử dụng chương trình, tương ứng với từng yêu cầu.

4. **Xử lý gian lận:**

- Sinh viên phải tự giác hoàn thành bài tập lớn bằng chính khả năng của mình. Sinh viên sẽ bị kết luận là gian lận khi có sự giống nhau bất thường giữa mã nguồn của các bài nộp. Trong trường hợp này, tất cả các bài nộp giống nhau đó đều bị kết luận là gian lận. Vì vậy Sinh viên phải bảo vệ mã nguồn bài tập lớn của mình.
- Sinh viên đặt tên sai hoặc không thực hiện đúng quy định nộp bài gây khó khăn trong quá trình chấm bài.
- Sinh viên không hiểu mã nguồn do chính mình viết.

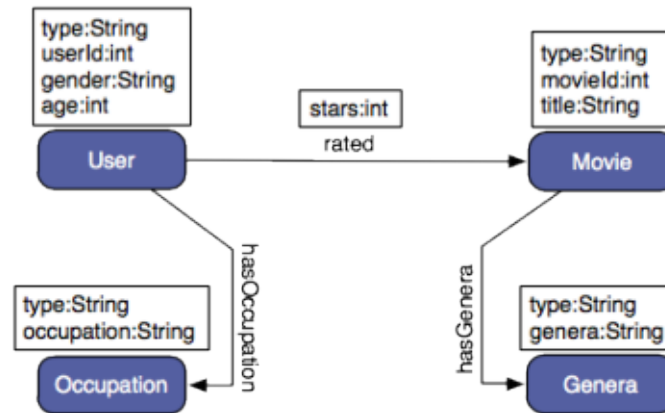
- Sinh viên không nộp bài hoặc bị nhận điểm 0

Sau mỗi bài tập lớn được nộp, một số sinh viên sẽ được gọi vấn đáp ngẫu nhiên để chứng minh rằng bài tập lớn đã được nộp là do chính mình làm. Nếu vi phạm các quy định trên sinh viên sẽ bị **CẤM THI CUỐI KỲ**

3 Nội dung

3.1 Phát sinh đồ thị (MovieRating)

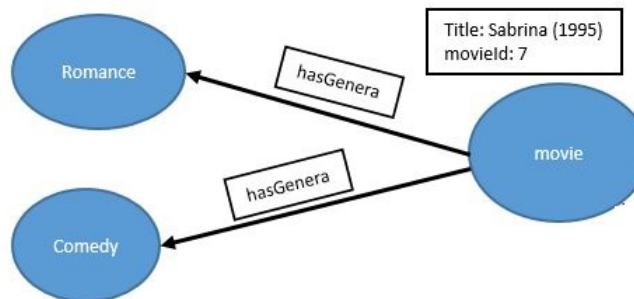
Tập dữ liệu **MovieLens** gồm 3 tập tin *user.dat*, *movies.dat*, *rating.dat*. Sinh viên cấu trúc lại dữ liệu theo lược đồ được cung cấp như hình bên dưới.



Để biết thông tin về tập dữ liệu, Sinh viên xem trong tập tin *Readme.txt*

Ví dụ

Trong tập tin *movies.dat*
 7::Sabrina (1995)::Comedy/Romance



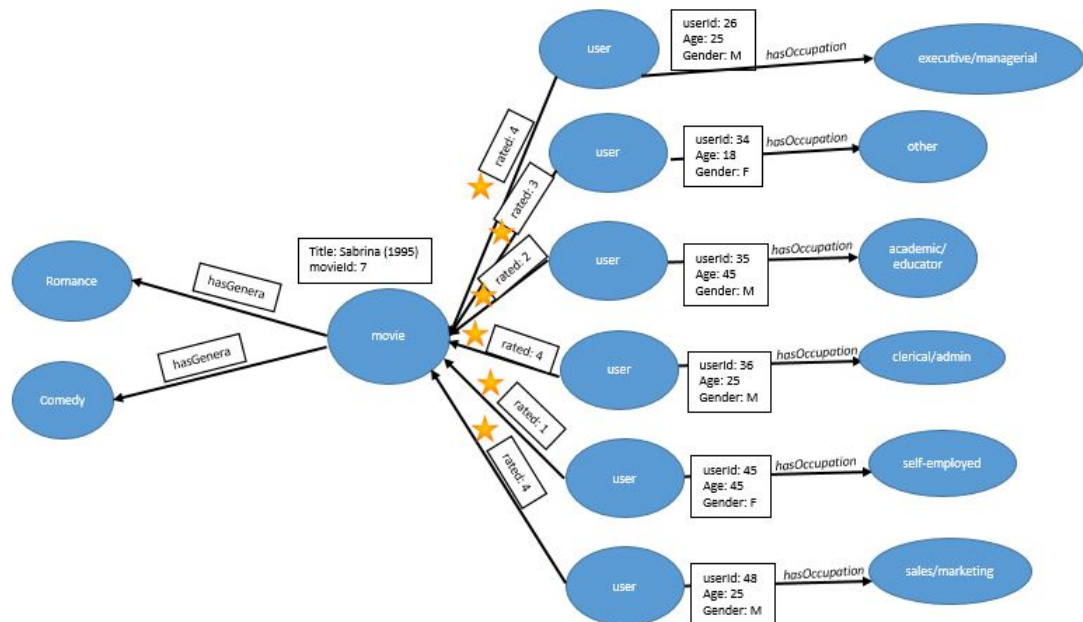
Trong tập tin *rating.dat* giả sử thu được các thông tin liên quan đến bộ phim Sabrina (1995) như sau:

26::7::4::978141783
34::7::3::978104490
35::7::2::978104122
36::7::4::1012707697
45::7::1::977989009
48::7::4::978013413

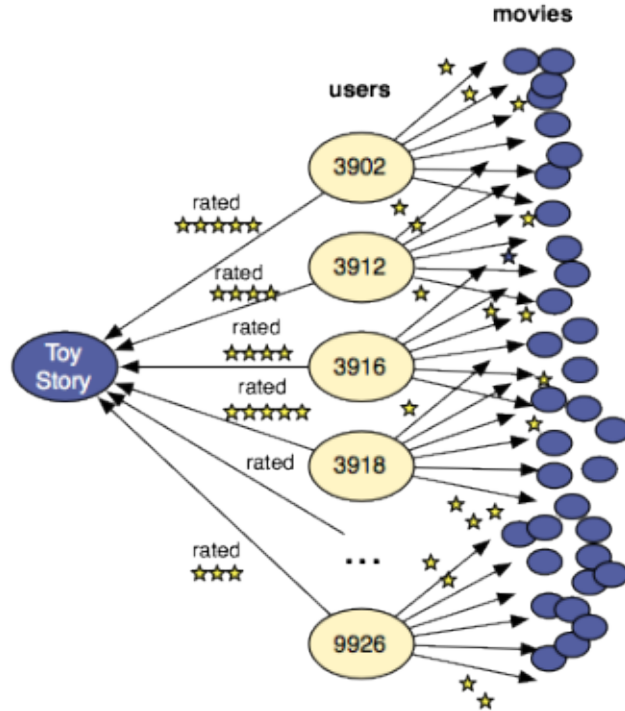
và *user.dat*

26::M::25::7::23112
34::F::18::0::02135
35::M::45::1::02482
36::M::25::3::94123
45::F::45::16::94110
48::M::25::4::92107

Đồ thị được biểu diễn lại như hình bên dưới



3.2 Duyệt đồ thị



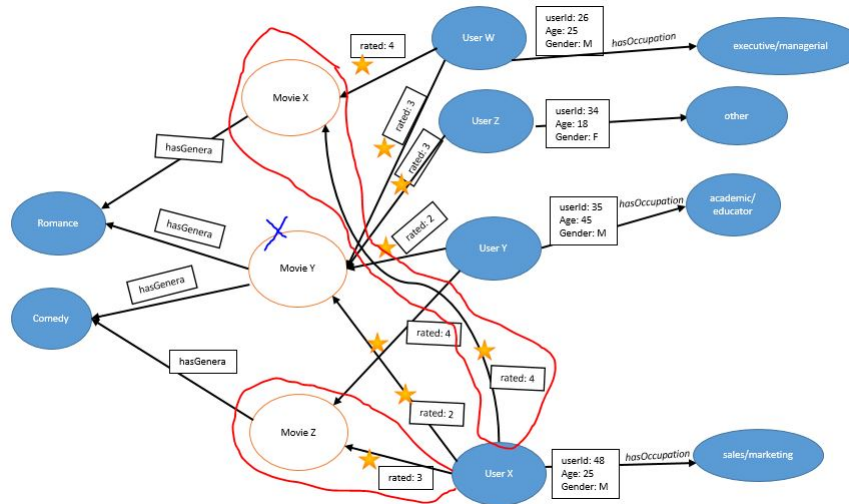
Yêu cầu 1: Xây dựng đồ thị $G = (V, E)$ từ tập dữ liệu **MovieLens**, với V là tập đỉnh và E là tập cạnh tương ứng của đồ thị G vào máy tính. **(3 điểm)**

Yêu cầu 2: Dựa trên đồ thị G . Sinh viên duyệt đồ thị theo những yêu cầu sau:

1. Cho biết những bộ phim được người sử dụng **X** đánh giá từ 3 sao. **(1 điểm)** (*Lưu ý:* Sinh viên viết hàm gồm 1 tham số **X**)

function GR_Req1

- **Input:** **X** là id của người dùng.
- **Output:** Danh sách các đỉnh Movies thỏa điều kiện.
- **Ví dụ:**

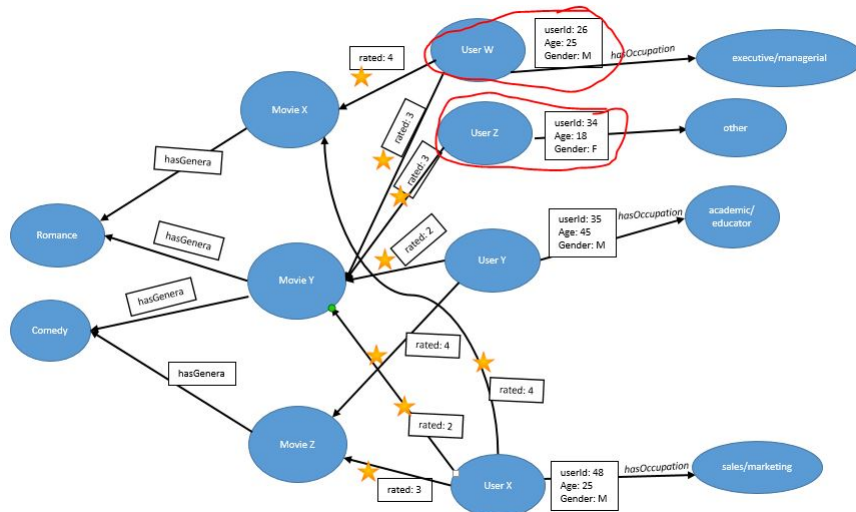


Danh sách những bộ phim được người dùng **X** đánh giá gồm: $\{Movie\ X, Movie\ Y, Movie\ Z\}$. Tuy nhiên danh sách kết quả trả về chỉ có $\{Movie\ X, Movie\ Z\}$ vì *Movie Y* chỉ được đánh giá 2 sao.

2. Cho biết những người sử dụng nào đánh giá bộ phim có tiêu đề **Y** từ 3 sao? **(1 điểm)**
(Lưu ý: Sinh viên viết hàm gồm 1 tham số **Y**)

function GR_Req2

- **Input:** **Y** là id của bộ phim
- **Output:** Danh sách các đỉnh Users thỏa điều kiện.
- **Ví dụ**



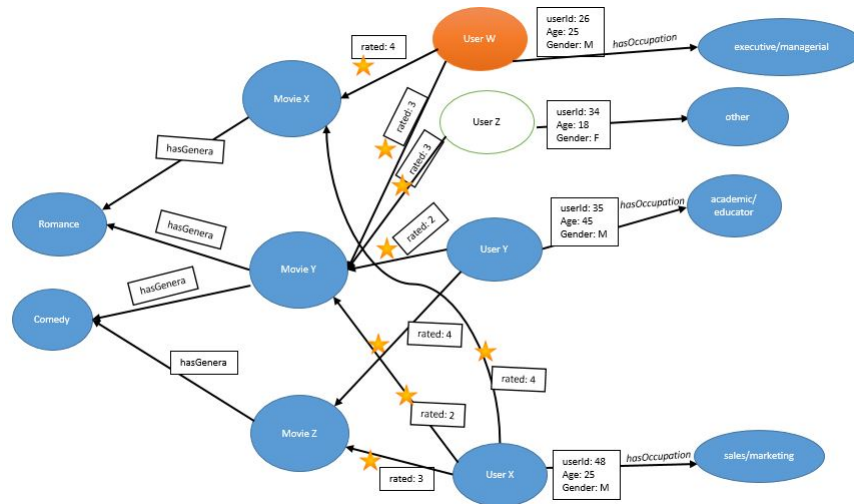
Danh sách những người dùng đánh giá bộ phim **Y** gồm $\{User X, User Y, User Z, User W\}$, trong đó chỉ có $\{User W, User Z\}$ đánh giá bộ phim **Y** từ 3 sao.

3. Cho biết những người sử dụng nào đánh giá bộ phim có tiêu đề **Y** trên 3 sao và đồng thời có ít nhất một bộ phim khác họ cũng đánh giá trên 3 sao? **(1 điểm)**

(Lưu ý: Sinh viên viết hàm gồm 1 tham số **Y**)

function GR_Req3

- **Input:** **Y** là id của bộ phim
- **Output:** Danh sách các đỉnh Users thỏa điều kiện.
- **Ví dụ**



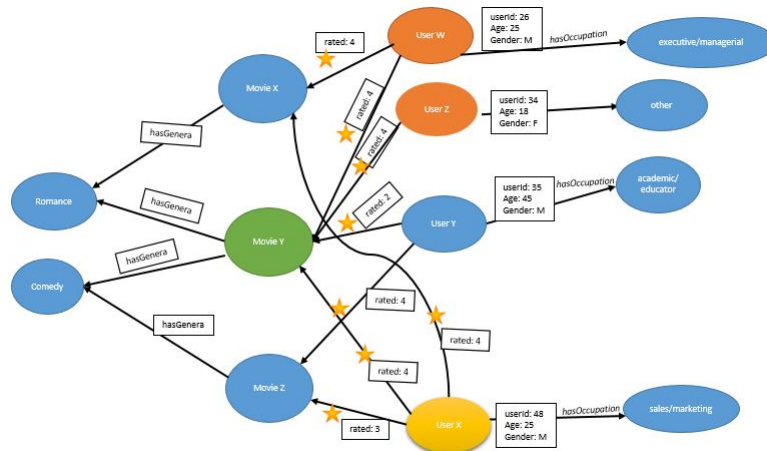
Danh sách những người dùng đánh giá bộ phim **Y** trên 3 sao {*User W*, *User Z*}, tuy nhiên kết quả trả về chỉ có {*User W*} vì ngoài bộ phim **Y** người dùng *User Z* không đánh giá bộ phim nào khác trên 2 sao.

4. Người dùng **X** đánh giá bộ phim **Y** 4 sao, tìm tất cả những người dùng nào có cùng nhóm tuổi với người dùng **X** và cũng đánh giá bộ phim **Y** từ 4 sao trở lên. (1 điểm)

(Lưu ý: Sinh viên viết hàm gồm 2 tham số **X**, **Y**)

function GR_Req4

- **Input:** **X** id của người dùng; **Y** là id của bộ phim
- **Output:** Danh sách các đỉnh Users thỏa điều kiện.
- **Ví dụ:**

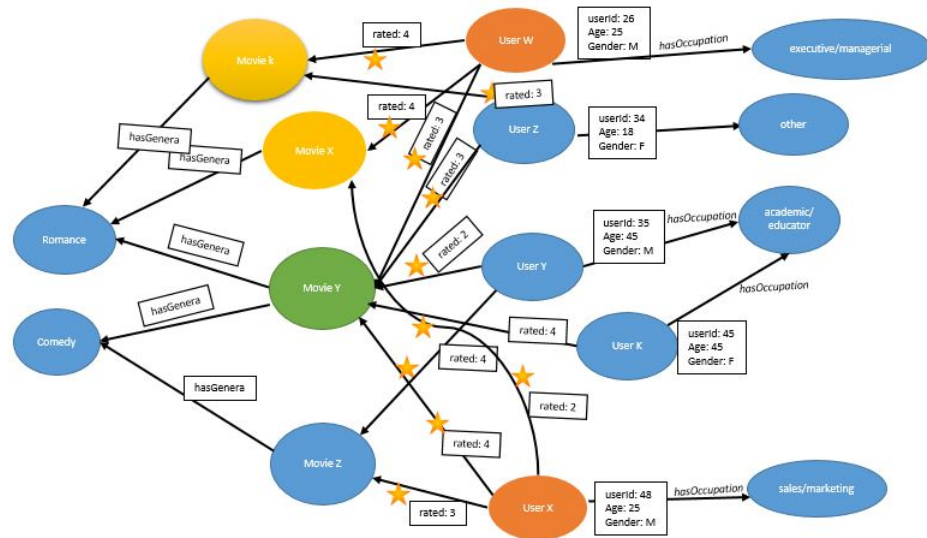


Người dùng **X** (màu vàng) đánh giá bộ phim **Y** (màu xanh lá cây) 4 sao. Từ bộ phim **Y** ta tìm ra tất cả những Người dùng (màu cam) của đồ thị có đánh giá bộ phim **Y** cũng từ 4 sao trở lên. Kết quả thu được là người dùng $\{User\ W\}$ vì người dùng *User Z* không cùng nhóm tuổi với người dùng **X**.

5. Hãy tìm ra 15 bộ phim hay nhất cùng thể loại với bộ phim **Y** được người dùng **X** và những người dùng khác cùng giới tính với người dùng **X** đánh giá từ 3 sao trở lên. **(1 điểm)**
(Lưu ý: Sinh viên viết hàm gồm 2 tham số **X**, **Y**)

function GR_Req5

- **Input:** **X** id của người dùng; **Y** là id của bộ phim
- **Output:** Danh sách 15 đỉnh Movies thỏa điều kiện.
- **Ví dụ:**



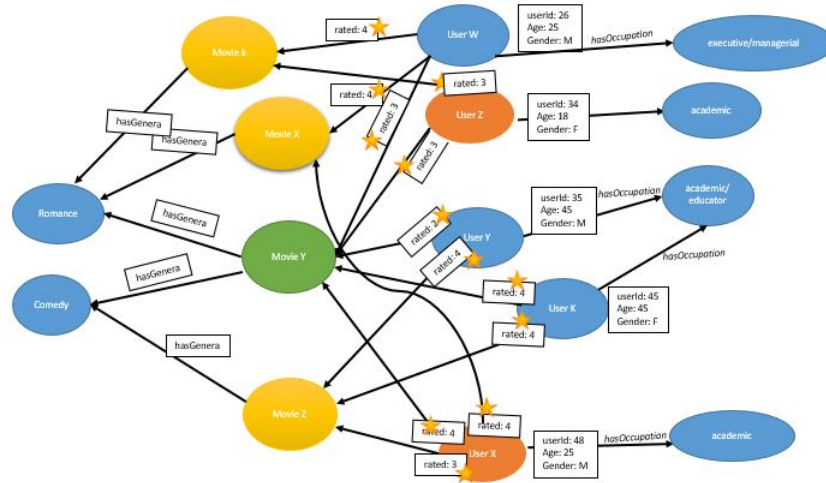
Những *User Z*, *User W*, *User K* cùng đánh giá bộ phim **Y** 3 sao như *User X*, nhưng chỉ có *User W* cùng giới tính với *User X* từ đó tìm ra được top những bộ phim cùng thể loại với bộ phim **Y** được *User W* đánh giá từ 3 sao. Giả sử trong trường hợp đồ thị bên trên ta thu được kết quả {*Movie X: 1*, *Movie K: 1*}. Danh sách phim được sắp xếp dựa vào chỉ số **score**, được tính từ tổng số cạnh đầu vào nối với đỉnh *Movie* có cùng thể loại với phim **Y** và chỉ tính những cạnh có chỉ số đánh giá từ 3 sao trở lên xuất phát từ những người dùng cùng giới tính với **X**.

6. Hãy tìm ra 10 bộ phim hay nhất cùng thể loại với bộ phim **Y** được người dùng **X** và những người dùng khác cùng nghề nghiệp với người dùng **X** đánh giá từ 3 sao trở lên (**1 điểm**)

(Lưu ý: Sinh viên viết hàm gồm 2 tham số **X**, **Y**)

function GR_Req6

- **Input:** **X** id của người dùng; **Y** là id của bộ phim
- **Output:** Danh sách 10 đỉnh *Movies* thỏa điều kiện.
- **Ví dụ:**



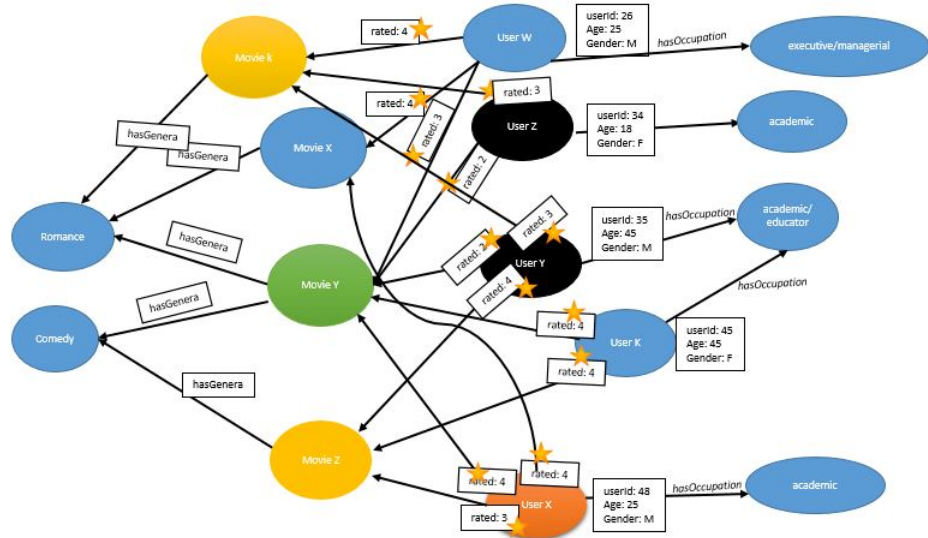
User W, User Z, User K cùng đánh giá bộ phim Y từ 3 sao trở lên như người dùng X. Tuy nhiên chỉ có User Z và User K cùng nghề nghiệp với người dùng X. Từ đó tìm ra được top 10 những bộ phim cùng thể loại với phim Y được User K, User Z đánh giá từ 3 sao trở lên. Giả sử trong trường hợp đồ thị bên trên ta thu được kết quả {Movie Z: 3, Movie X: 1, Movie K: 1}. Danh sách phim được sắp xếp dựa vào chỉ số **score**, được tính từ tổng số cạnh đầu vào nối với đỉnh Movie có cùng thể loại với phim Y và chỉ tính những cạnh có chỉ số đánh giá từ 3 sao trở lên xuất phát từ những người dùng cùng nghề nghiệp với X.

7. Hãy tìm ra 5 bộ phim được yêu thích nhất bởi những người dùng không thích bộ phim Y mà người sử dụng X thích. **Quy ước:** những bộ phim được xem là được yêu thích nếu bộ phim đó được đánh giá từ 3 sao trở lên. (1 điểm)

(Lưu ý: Sinh viên viết hàm gồm 2 tham số X, Y)

function GR_Req7

- **Input:** X id của người dùng; Y là id của bộ phim
- **Output:** Danh sách 5 đỉnh Movies thỏa điều kiện.
- **Ví dụ:**



$\{User Y, User Z\}$ không thích bộ phim **Y** mà người dùng **X** thích, từ đó ta tìm ra được danh sách những bộ phim được $\{User Y, User Z\}$ thích. Giả sử trong trường hợp đồ thị bên trên ta thu được kết quả $\{Movie K: 2, Movie Z: 1\}$. Danh sách phim được sắp xếp dựa vào chỉ số **score**, được tính từ tổng số cạnh đầu vào nối với đỉnh Movie và chỉ tính những cạnh có chỉ số đánh giá từ 3 sao trở lên xuất phát từ những người dùng không thích phim **Y**.

Kết quả lưu trữ theo từng tập tin *MSSV_HoTen_NhomX_ReqY*

Ví dụ: Sinh viên có tên Nguyễn Anh Huy có mã số sinh viên 51403361 học lý thuyết nhóm 1 và lưu nội dung kết quả câu 1 trong Yêu cầu 2.

Tập tin kết quả output phải được đặt: *51403361_NguyenAnhHuy_Nhom1_Req1*, tương tự lưu nội dung kết quả câu 2 trong Yêu cầu 2: *51403361_NguyenAnhHuy_Nhom1_Req2*