설치: https://gephi.org/ 에서 Gephi를 설치하면 됨.
단, JDK(Java Development Kit)가 안 깔려 있을 경우,
http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
에서 JDk를 먼저 설치하고 Gephi를 설치해야함.

제출: 2019년 11월 25일 월요일 오전 11시 59분 59초

주의 사항: Late Submission 없음.

==========

1. Adjnoun : 기초 Gephi 실습

소설 David Copperfield에서 자주 나오는 형용사 명사 이웃관계를 나타낸 그래프를 이용한 실습이다.

자세한 내용 및 Data sets 출처 :
http://www-personal.umich.edu/~mejn/netdata/
[필요한 File]
adjnoun.gml : 형용사 - 명사 인접 관계를 명시한 파일로, label은 단어(Spelling)를 의미하며, value가 0일 경우 형용사, 1일 경우 명사를 의미함.
==========

1) 해당 파일을 불러 다음을 만족하는 그래프를 만드시오.
   (1) Label을 출력하시오.
   (2) Force Atlas Repulsion strength 1000 Attraction을 1 줘서 그래프를 만드시오. (옵션을 Adjust by Sizes를 줘서 모든 라벨이 겹치지 않게 하시오.)
   (3) 형용사와 명사가 다른 색으로 표시되게 출력하시오.
   (4) Network Diameter의 통계를 이용하여 Betweenness Centrality(Network Diameter)에 따라 노드의 크기를 변경하시오(Min : 10, Max : 50)

위에 조건을 갖춘 그래프를 스크린샷으로 출력하여 보고서에 첨부할 것. 이 때, 간선과 노드가 제대로 모두 보이도록 하시오. 문제에 제시되지 않은 것은 모두 default값을 이용하시오.

2) 1)의 그래프를 이용하여 다음의 문제를 답하시오.
   (1) time 과 관련된 모든 **형용사**를 적으시오.
   (2) small 과 관련된 **명사**는 모두 몇 개 인가?
   (3) certain 과 관련된 **명사**를 모두 적으시오
   (4) 그래프에서 Betweenness Centrality가 높은 노드를 3개를 적으시오.

3) Modularity(default 이용)를 이용하여 Size Distribution 결과 Report(Results + Size

Distribution 그래프)와 **Modularity Class를 기준으로 노드의 색을 다시 구분한 그래프**를 출력하시오.

4) 3)의 그래프를 이용하여 다음의 문제들을 답하시오. [0.2점]

(1) thought와 연관되어 있는 단어들 중 같은 Modularity Class에 있는 단어들을 모두 적으시오.

(2) place와 연관되어 있는 단어들 중 같은 Modularity Class에 있는 단어들은 모두 몇 개 인가?

문제에서 언급하지 않은 다른 모든 조건들은 함수의 Default값을 사용하시오.

==========

2.Stack-overflow-tag-network : 분할된 파일을 합친 Gephi 실습

Stack Overflow의 기술들이 서로 얼마나 연관되어 있는지 알아보는 실습으로, tag correlation을 이용하여 기술의 사용 정도 및 서로간의 연간관계 알아보기 위한 실습이다.

자세한 내용 및 출처:
https://www.kaggle.com/stackoverflow/stack-overflow-tag-network
[필요한 file]
stack_network_links.csv : 간선과 관련된 정보가 저장되어 있음. Value는 해당 언어와 다른 언어의 연관도를 의미함.(클수록 더 연관이 높음)
stack_network_nodes.csv : 노드와 관련된 정보가 저장되어 있음. Nodesize는 쓰이는 정도를 의미함.(클수록 더 많이 씀)
==========

※참고 : Links 파일은 Edges table로 import할 것.

1) 다음의 조건을 만족하는 그래프를 작성하시오.

전처리:

Links.csv파일을 먼저 import하고 nodes.csv파일을 import하시오.
또한, nodes.csv을 import 받되, New workspace가 아닌 Append to existing workspace로 설정해 같은 workspace안에 그려지게 하시오.

Gephi:

(1) 노드들의 이름들이 출력되게 하시오.(여기서 Label은 Id와 같다.)
(2) Label들의 크기가 Nodesize에 맞춰서 크기가 설정되게 하고 fontsize scale을 설정하여 글씨가 너무 크게 나오지 않게 하시오.(Min: 1, Max: 30)
(3) group에 따라 Node의 색을 달리하고, nodesize column에 따라 Node의 크기를 달리하시오. (Minimum:1, Maximum size:50)
(4) 간선의 색은 단색으로 하고, 크기를 Weight scale에 따라 조정하시오.(이 때, 전처리가 잘 되었다면, 간선의 크기는 Weight에 따라 굵기가 달라야 한다.)
(5) Force Atlas(Repulsion 500, Attraction 1)을 이용하여 그래프를 만드시오.

* 조그마한 그래프들이 잘리지 않게 할 것.

2) 1)의 그래프를 이용하여 다음의 문제들을 답하시오.

* Node size는 단어의 쓰임정도를 뜻한다.

(1) 가장 많이 쓰는 언어 3개를 순서대로 적으시오.

(2) 연결 그래프는 총 몇 개인가?

(3) 다음 언어와 연관된 언어 중 가장 많이 쓰이는 언어를 적으시오

(3-1) html

(3-2) Machine-Learning

(3-3) Hadoop

(3-4) Linux

(3-5) Git

문제에서 언급하지 않은 다른 모든 조건들은 함수의 Default값을 사용하시오.

==========

3. Bitcoinalpha : Filtering과 Dynamic Graph 실습

Bitcoin을 이용한 플랫폼인 Bitcoin Alpha에서의 who-trust-whom network 그래프이다. Bitcoin 유저는 모두 익명이기 때문에, 유저의 신뢰도가 매우 중요하다. 따라서, 이 실습에서는 유저가 다른 유저에게 준 신뢰도를 이용하여 실습이다.

자세한 내용 및 출처:
http://snap.stanford.edu/data/soc-sign-bitcoinalpha.html
[필요한 file]
bitcoinalpha.csv: Bitcoin Alpha를 이용한 who-trusts-whom network 파일 {Source, Target, Weight, Time}으로 정해져 있으며, Weight의 경우 Source가 Target에 대한 평가가 -10부터 10까지 정해져 있음. Time은 평가한 시간이 적혀져 있음.

==========

1) 다음의 조건을 만족하는 그래프를 작성하시오.

Gephi:

(1) 전처리된 파일을 Gephi에서 import하시오.
Edges table로 import하며, **시간은 String으로 받을 것.+Time representation을 Timestamps가 아닌 Interval로 할 것.**

(2) Data Laboratory의 Edge tab에서 Merge Column을 선택하여 time과 interval을 merge하되 Create time interval을 이용하여 Timestamp를 만드시오.(Default end time에 2016-01-31로 할 것(Parse Dates))

(3) Edge를 Weight에 따라 다르게 색을 지정할 것(Ranking이용, 확실한 구분을 위해 음수 쪽은 빨간색 양수는 파란색, 0에 가까울수록 흰색을 하는 것을 추천함). 또한, Degree 크기에 따라 Node의 크기를 바꿀 것.(Min: 10, Max: 50, 색도 바꾸는 것을 추천)

(4) Label을 넣을 것.(여기서 label은 ID와 같음)

(5) Fruchterman Reingold를 이용하여 그래프를 만드시오.(자료가 매우커서 시간이 오래 걸리므로 speed를 올리는 것을 추천. 어느 정도 됐다고 생각하면 중지하고 다음 단계를 하시오.)

(6) **Custom Time Bounds & interval**을 이용하여 **2010년 11월 7일부터 11월 25일 까지**의 그래프만을 보고서에 첨부하시오(단 Node까지 지울 필요는 없음, 만약 해당 그래프가 잘 모여 있지 않는다면 Fruchterman Reingold를 다시 하여 뭉쳐진 그래프로 첨부할 것.)

- Custom Time Bounds & Interval: Filters > Dynamics > TimeInterval 더블클릭

- 우측 하단에 OpenTimeline클릭 > 좌측 하단에 톱니바퀴 클릭 > Set time format에서 Date 클릭 > Ok 클릭

- 좌측 하단에 톱니바퀴 클릭 > Set custom bounds클릭 > Interval 부분에 문제의 조건 날짜 입력 > Ok 클릭 > 우측에 filter 클릭

2) **1)-(5)의 그래프에서 다음의 조건을 추가로 만족하는 그래프를 작성하고 다음 문제를 푸시오.**

(7) filter를 이용하여 Weight가 0 초과인 edge들을 모두 제거하시오.

(8) PageRank에 따라 Node의 크기를 바꾸시오.(Max: 50)(또한, 사이즈에 따라 글자 label크기도 바꿀 것, Max: 5)

(9) Fruchterman Reingold를 한 번 더 하여 그래프를 정리하여 붙임하고 다음의 문제를 푸시오.

(2-1) PageRank는 그 그래프에서 상대적 중요도를 의미한다. 이를 이용하여, 악성 사용자를 찾아내려고 한다. **악성 사용자일 확률이 높은 노드 5개**를 찾으시오

(2-2) 다시 Degree에 대한 통계를 만든 후, In-Degree를 이용하여, 위의 5개의 노드 중 **가장 악성 사용자일 확률이 높은 노드 하나**와 **가장 악성 사용자가 아닐 확률이 높은 노드 하나**를 찾으시오.

(2-3) ID 117이 부정적 평가를 내린 Node 중 한 노드는 117 노드를 부정적 평가를 내렸다. 이 노드를 적으시오

(2-4) 1번 노드의 **PageRank**를 찾으시오.

문제에서 언급하지 않은 다른 모든 조건들은 함수의 Default값을 사용하시오.

==========

4. Facebook_Tvshow : 전체 실습

페이스북 페이지에 관련된 그래프로, 여기서 사용될 페이지는 티비쇼와 관련된 검증된 페이스북 페이지이다. 각 노드들은 익명성이 보장되어 있으며, 각 간선들은 서로간의 “mutual likes”이다. 이 실습에서는 위의 자료를 이용하고, 위에서 연습했던 기능을 이용하여 Social Network관련 그래프 작성 및 자료 활용에 대한 전반적인 능력을 실습한다.

자세한 내용 및 출처:
http://snap.stanford.edu/data/gemsec_facebook_dataset.html
[필요한 file]
tvshow_edges.csv

==========

1) 다음의 조건을 만족하는 그래프를 작성하시오.

(1) import 된 graph는 undirected일 것.
(2) Node의 크기는 Betweenness Centrality 따른 크기로 정할 것.(최대 50)
(3) Node 크기에 따른 라벨을 크기를 설정할 것
(4) Label을 넣을 것.(여기서 label은 ID와 같음)
(5) Clustering Coefficient가 0.8이상인 것은 제외하고, Modularity Class의 Partition 개수가 100 이하인 Node들을 제외할 것
(6) Modularity class에 따라 Node의 색을 결정할 것.
(7) ForceAtlas 2를 이용할 것. (scaling 1.5, Prevent overlap)
데이터가 많이 Preview가 안 나올 수 있으므로, 안 나올 경우 Overview의 스크린 샷을 찍어서 제출해도 됨.

2) 1)의 그래프를 이용하여 다음 문제를 답하시오.

(2-1) 1)의 그래프의 **새로운 Network Diameter**를 구하시오.
(2-2) 위의 그래프에서 **가장 영향력이 높은 노드 5개**를 순서대로 나열하시오.
(2-3) Node 2589에서 3586까지 **Shortest path length**를 찾아 경로를 쓰시오.
(2-4) Node 299와 연관된 노드 중 **가장 영향력이 큰 노드**를 적으시오.

문제에서 언급하지 않은 다른 모든 조건들은 함수의 Default값을 사용하시오.