Report for the project

2011120322 홍성민

# What data structure you chose and why.

우선 이 프로젝트는 rep파이썬에서 구현하였으며, 처음에 text 파일을 받아들이는 과정에서는 line 별로 슬라이싱을 한 후 각 사용자 별 정보를 node로 구성하였습니다. 그리고 각각의 node를 list에 저장하여, userprofilelist, userfriendship, tweets 의 이름으로 저장하였습니다.

하지만 text 파일의 정보에서는 위 프로젝트를 실시하는데 필요없는 id에 대한 정보나 날짜에 대한 정보가 있기 때문에, 좀 더 효율적인 자료구조 구현을 위해, 필요한 정보들을 뽑아 hash table을 만들었습니다. Hashing을 선택한 이유는 hashing이 시간복잡도가 가장 작고( O(1)), 제시된 문제해결을 위해 가장 효율적이라고 생각했기 때문입니다.

# What is your expected performance?

먼저 버튼 0번을 눌러 파일을 불러들이면, raw data에 대한 데이터 (total tweets, total friendship records, and total users)가 표시되고, 그 이후에는 메뉴의 버튼을 누르고 나오는 지시사항을 따라 진행하면 문제없이 프로그램이 작동합니다. 다만, 시간관계상 scc와 shortest path의 경우, 각각 관련 정보를 vertex에 저장하고 dfs의 transpose의 방법을 통해 scc를 구하고, shortest path는 dijsktra를 이용하는 것까지는 생각했고, 관련 코드도 포함했지만 구현하지 못했습니다.

# How would you improve the system in the future?

우선 완료하지 못한 extra-credit 요소들을 완료하고, 만일 이 프로젝트가 확장되어도 raw data의 쓰이지 않는 파일이 계속해서 쓰이지 않는다면, text파일을 받아들일때 바로 hashing의 형태로 저장하는 방법을 고안할 것입니다.