

# Twitter Project

컴퓨터 2015410041 홍주연

## 1. What data structure you chose and why

### 1) stack

-전체적인 알고리즘에 stack을 이용해서 list를 만든다. 6, 7번 메뉴에서 user 정보와 tweet을 삭제할 때 pop과 remove를 이용한다.

### 2) directed graph

-8, 9번 메뉴에서 directed graph와 weight를 이용해서 strongly connected components와 shortest path를 찾는다.

3) 나머지 메뉴에서는 for, if문을 사용해서 기능을 수행한다.

## 2. What is your expected performance

### 0) Read data files

-각 파일의 각 줄을 list에 저장해서 그 list의 총 길이를 이용해서 total users, total friendship records, total tweets를 계산한다.

### 1) display statistics

-tweetcount와 friendcount를 정의해서 friend 수와 tweet 수가 가장 적은 경우와 가장 많은 경우를 계산하고 평균을 계산한다.

### 2) Top 5 most tweeted words

-tweet의 개수를 array에 저장하고 하나하나 비교하면서 가장 높은 숫자 5개를 출력한다.

### 3) Top 5 most tweeted users

-2번과 비슷한 방식으로 배열 tcount의 값을 비교해가면서 가장 트윗 수가 많은 사용자 5명을 순서대로 출력한다.

### 4) Find users who tweeted a word

-4번을 선택하면 프로그램 사용자가 찾고자 하는 단어를 입력하게 하고, 그 단어를 전체 tweet에서 찾아 그 단어를 사용한 user를 차례대로 출력한다.

5), 6), 7)은 4번을 실행한 후에 할 수 있다.

### 5) Find all people who are friends of the above users

-4번을 실행할 때 user 수만큼의 배열을 새로 만들어서 그 배열의 특정 단어를 사용한 user의 인덱스에 1을 저장한다. 만약 그 배열의 element 값이 1이면 friend 정보를 백업해둔 friend 배열에서 ""(empty)가 아니면 출력한다.

### 6) Delete all mentions of a word

-그 단어가 등장한 인덱스를 저장하는 tindex1 배열을 새로 만들어서 값이 1이면 tweetlines에서 그 단어가 등장한 block을 삭제한다. Index out of range를 피하기 위해 count6 변수를 지정한다.

### 7) Delete all users who mentioned a word

-tweet 배열에서 단어가 나오면 identification number, sign-up date, screen name을 저장한 배열과 userlines 배열에서 그 user의 정보를 삭제한다.

### 99) Quit

-프로그램을 종료한다

0, 1, 2, 3, 4, 5, 6, 7, 8, 99가 아닌 다른 string을 입력하면 다시 입력하도록 반복문을 돌린다.

### 3. How would you improve the system in the future

-코드를 간략하게 해서 시간복잡도를 줄여 실행 속도를 줄인다. red black tree, Dijkstra algorithm 등을 사용해서 더 읽기 쉽고 빠르게 수행하는 알고리즘을 작성한다.