계획

User profile

유저 프로필은 연결 리스트로 관리하려고 한다. 유저 프로필을 포인터를 이용한 연결 리스트로 관리할 경우 유저 삭제 기능을 쉽게 구현할 수 있다.

Friendship

친구 관계는 연결 리스트로 관리하려고 한다. 그리고 이는 인접 리스트라 하며 친구 관계는 undirected하므로 (A, B)가 입력되면 A에서 B로 B에서 A로 둘다 추가를 해주면 된다.

Word tweet

이 역시 연결 리스트로 관리하는데 포인터를 이용해서 유저 프로필과 이어준다.

• Advanced search – Which word is tweeted the most by all users

– Which user tweeted the most words

->선형 탐색 알고리즘을 사용한다. O(n)의 시간 복잡도를 가지는데 친구를 많이 가진 순서대로 딱히 정렬된 것이 아니므로 일반적인 선형 탐색 알고리즘을 선택했다.

– Find users who tweeted a word (e.g., ’연세대’)

->이 역시 위와 같은 이유로 선형 탐색 알고리즘을 사용한다.

– Find all people who are friends of at least one of the above found users

->위 두 가지 사람들을 스택으로 push하고 pop을 할 때 겹치는 사람은 출력하지 않는 방식으로 구현한다. 시간 복잡도는 O(n)이다.

• Advanced delete

– Delete all mentions of a word

->해당 글을 찾은 후 삭제한다. 해당하는 글을 찾는 알고리즘은 역시 정렬이 된 상태가 아니므로 선형 탐색 알고리즘을 쓸 것이다.

– Delete all users who mentioned a word ∗ This requires updating friendship database • Graph algorithms

->우선 해당 글을 찾은 후 그 글을 쓴 유저의 모든 글을 삭제한 후 해당 유저의 프로필을 삭제할 것이다. 해당 글을 쓴 유저를 찾는 알고리즘은 역시 선형 탐색 알고리즘을 쓸 것이다.

– Top 5 Strongly connected components (most number of users)

-> Kosaraju's **algorithm을 사용할 것이며 시간 복잡도는 O(V+E)이다.**

– Top 5 Shortest path (weight = number of friends)

->undirected 이므로 BFS를 이용해서 첫번째 유저에서 두번째 유저까지의 최소 거리를 구할 수 있다.