Project Report

2015410054 이강빈

1. Development enviroment

Visual Studio 2015, Visual C++

2. What data structure you chose and why

단어와 유저를 저장하는데 사용한 자료구조는 Red-Black Tree이며 각각의 단어에는 트윗한 유저를 저장하는 Linked List를 각각의 유저에는 그 유저의 친구(Following)와 그 유저를 Follow하고 있는 유저, 그리고 그 유저가 트윗한 단어를 Linked List를 사용하여 저장하였다. 단어와 유저를 저장하는데 Red-Black Tree를 사용한 이유는 입력이 정렬되어 들어오는 경우에 BST는 입력, 검색, 삭제가 O(n)에 이루어지기 때문에 너무 느리다고 생각해 입력, 검색, 삭제를 O(logn)만에 할 수 있는 Red-Black Tree를 사용하기로 결정했다. 트윗한 유저나 유저의 Following, Follower를 저장하는데 Linked List를 사용한 이유는 배열을 사용할 수도 있지만 배열을 사용했을 경우 배열의 최대 크기보다 더 많은 입력이 들어왔을 때 저장을 할 수 없기 때문에 동적할당을 할 수 있는 Linked List를 사용하였다.

3. What is your expected performance

단어와 유저를 입력 받아 트리에 저장하는 것은 Red-Black Tree로 구현하였기 때문에 n개의 입력이 들어왔을 때 O(nlogn)의 성능을 보여줄 것으로 예상되고 친구수가 가장 많은(적은) 유저와 트윗을 가장 많이(적게)한 유저를 찾는데에는 BFS를 사용하여 트리를 탐색했기 때문에 O(n)만큼의 시간이 걸릴 것으로 예상된다. Top 5 most tweeted words와 Top 5 most tweeted users도 BFS를 사용하여 트리를 탐색하였기 때문에 O(n)(5n)의 시간이 걸릴 것으로 생각된다. 어떤 트윗을 한 유저를 찾는데에는 트윗한 단어 안에 그 트윗을 한 유저들을 모두 Linked List로 저장되어 있으므로 트리에서 그 단어를 찾는 시간 O(logn)의 시간이 걸릴 것이며, 3번을 실행했을 때 출력했던 유저들을 배열에 저장했기 때문에 3번에서 출력된 유저의 친구들을 출력하는 작업은 O(1)(유저들의 친구 수 만큼의 시간이 걸릴 것이므로)의 시간이 걸릴 것으로 예상되며 4번 또한 4번의 단어를 찾아 저장했기 때문에 O(1)의 시간이 걸릴 것으로 생각된다. 어떤 단어를 입력 받아 그 단어를 모두 삭제하는 작업은 Tree에서 단어를 찾고 그 단어를 트윗한 유저를 찾아가 유저가 트윗한 단어를 저장한 Linked List에서 그 단어를 삭제해야 하는 작업이 필요하지만 트윗한 유저를 찾고 Linked List에서 찾아 삭제하는 작업은 데체적으로 Constant Time이 걸릴 것으로 예상되므로 단어를 삭제하는 작업은 Tree에서 단어를 찾는 시간인 O(logn) + α정도의 시간이 걸릴 것 같다. 어떤 단어를 트윗한 모든 유저를 삭제하는 작업은 Tree에서 단어를 찾고 그 단어를 트윗한 모든 유저를 찾은 다음 그 유저를 삭제하는 작업을 해야하는데 유저를 삭제하는 작업이 유저의 Following 목록에 있는 유저를 찾아가 찾아간 유저들의 Follower목록에서 삭제하려는 유저를 지우고 Follower목록에 있는 유저를 찾아가 찾아간 유저들의 Following목록에서 삭제하려는 유저를 지우고 지우려는 유저가 트윗한 모든 트윗을 찾아가 그 트윗에 있는 유저목록에서 지우려는 유저를 삭제해야 하기 때문에 유저를 삭제하는데 O(n)정도의 시간이 걸릴 것으로 생각된다. 따라서 어떤 단어를 트윗한 모든 유저를 삭제하는 작업은 유저를 지우는 작업(O(n))에 단어를 찾는 작업(O(logn))을 곱한 O(nlogn)정도의 시간이 걸릴 것 같다.

4. How would you improve the system in the future

단어와 유저에 Linked List로 저장했던 것들을 Red-Black Tree나 BST로 저장한다면 단어를 트리에 넣을 때 더 빠른 시간으로 넣을 수 있을 것이며 유저를 삭제하는 작업에서도 Linked List로 탐색하는 것 보다는 빠를 것이기 때문에 그 부분을 수정한다면 더 빠른 시간으로 작업을 수행할 수 있을 것 같다.