2015410062 신기철

자료구조 project

1. STL 을 적극 활용, 코딩에 편의를 도모하였습니다.

2. 거의 모든 정보들을 vector를 활용해 저장했습니다.

3. STL에 관한 정보들을 많이 익혔습니다

저의 작업물에 대한 퍼포먼스는 다음과 같을 것입니다.

input load 시 :

user load -> O(N) : 그대로 입력을 받았습니다.

user load 후 -> sort를 통해 renumbering 하여 나중에 access 할 때 binary search로 log n 만에 access할 수 있도록 했습니다. sort 는 selection sort 를 이용해 O(N^2), binary search는 O(log N)으로, 직접 구현했습니다.

friend load -> E만큼의 edge를 각각 node에 대한 ID\_name에 대해 access를하는 과정을 수행하므로, O(E log N)이 수행됩니다. , set container를 많이 활용했습니다.

word load -> word들을 string으로 처리해 전부 load했습니다. 그리고 map container를 많이 활용했습니다.

action 시 :

0, 1 action은 O(1) , O(N)만큼 수행됩니다.

2, 3 action은 대략 O(N)으로, max를 5개씩 구합니다. 자료는 미리 map,vector 등에 저장되어있습니다. 물론 map의 access는 log N만큼을 더 곱합니다.

4 action은 word에 대한 map을 참조하고, word string에 대해 mapping되어있는 user들의 정보를 출력합니다. map을 사용해 access가 적게 걸립니다.

5 action은 above users에 대해 friend들을 탐색하므로 (above user의 수) \* (edge)만큼 수행됩니다.

6 action은 해당 word에 대해 mapping 된 user들의 tweet 값을 변경시키는 등 map 연산을 사용합니다.

7 action은 완벽하게 구현하진 못했습니다.

8 action은 Kosaraju의 방법을 따랐으며 O(E)만큼이 수행됩니다.

9 action은 Dijkstra's Shortest Path Algorithm을 O(E log N)의 시간복잡도를 같도록 직접 Heap을 구현하여 만들었습니다.