Yoon Jee Seok (윤지석)

2012210073

Data Structure class 2 – Prof. Kim Jong Kyu

2016-06-25

Term Project Report - Word Tweet

모든 data structure은 array list로 구현하였다.

여러모로 비효율적이지만 개발 시 편하였기 때문에 그냥 사용하였다.

1. Read data files

총 3가지의 data file을 불러오고 data structure에 저장한다. 따라서 insertion만 사용한다. 모두 unsorted array list를 사용하기 때문에 insertion 자체는 O(1)을 차지한다 (tweet는 sort 하기 때문에 O(n)). 하지만 User Profile 같은 경우 중복되는 id #이 있으면 안되기 때문에 O(n(+1))을 사용한다. 또한 friend 와 tweet은 id #이 user profile에 저장되어 있는지 확인하기 때문에 O(n(+1))을 사용한다.

사실 프로그램 자체로 insert 할 수 있는 기능이 없기 때문에 sorted data structure을 사용하는게 더욱 효율적이겠지만 function을 사용할 때 여러가지 key에 대해 sort 하기 때문에 이 파트에서는 unsorted data structure을 사용 하였다.

1. display statistics

Average, Min, Max를 구하기 위해선 모든 data를 봐야 하기 때문에 O(n)을 사용한다.

Array list가 아닌 sorted data structure을 사용하였다면 min, max 는 O(1)이 되 수 있기 때문에 미래에는 sorted data structure을 사용하는 것이 좋을 것 같다. 하지만 average를 구하는 경우는 어차피 O(n)을 사용하기 때문에 그냥 sort를 하지 않았다.

1. Top 5 most tweeted words/user

Content에 대해 sort를 한다(O(n)). Max를 찾는다 (O(n)). 즉 O(n+n)을 사용한다. Sort를 시행할 때 새로운 array list를 사용하기 때문에 space complexity는 x2가 된다.

Tweet은 id #에 대해 sorted array list를 사용하기 때문에 조금 더 효율적이지만 duplicate element가 많을수록 똑같이 O(n)이 나온다. Word Tweeter은 duplicate element가 많기 때문에 어쩔 수 없이 sorted나 unsorted나 성능은 비슷하다고 봐야한다.

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

Word search를 하기 때문에 모든 data를 비교하기 때문에 O(n)을 사용한다. Search algorithm은 array list를 사용하였기 때문에 말 그대로 index 0부터 n까지 1씩 increment 해가는 수동적인 방법을 사용하였다. 여기서 return 받은 id #을 다른 function에서 사용하기 한다.

조금 더 생각을 했다면 BST 나 RB tree를 사용하는 것이 좋았을 것 같다.

1. Find all people who are friends of the above users

User list에서 이름을 검색하고 friend가 있는지 검색하기 때문에 O(nlogn)을 사용한다.

이쯤 와서 깨달은 것은 hash table을 사용하는 방식을 채택해야 했다는 것이다. 처음엔 function 마다 다른 key를 사용하기 때문에 별 문제 없으리 생각했지만, 생각과 다르게 user id를 거의 모든 function에서 사용한다. Hash table을 사용하면 사실 성능은 비슷하겠지만 개발 속도가 몇배로 빨라졌을 것 같다.

1. Delete all mentions of a word/user

Friend, user, tweet 모두 array list를 사용하기 때문에 delete는 O(n)을 사용한다.

Array List를 사용하면서 좋은 점은 여기서 볼 수 있다. 편하다.

1. Find strongly connected components

DFS를 사용해 SCC를 구한다. 이 소스 코드는 교수님이 수업시간 때 사용하신 코드를 수정해서 만든 것 이다.

교수님 소스코드와 다른 것은 SCC를 모두 출력하는 것이 아닌 Top 5만 고르는 것이다.

1. Find shortest path from a given user

Dijkstra algorithm은 그래프를 사용해서 구현하였다. 알고리즘 자체는 교재에 나온 것과 동일하다.

개발 중점

개발 중 가장 중점으로 두었던 것은 개발 시 어떤 것이 가장 편할지 였다. 단지 편해서 선택한 것이 아닌 장점과 단점을 조합해본 결과 다른 data structure 및 algorithm을 사용하는 것보다 개발을 가장 효율적으로 할 수 있는 방법을 채택하였다. 따라서 많이 부족해 보일 수 도 있지만 나쁘지 않은 결과를 내 주어 나쁘지 않다고 생각한다.

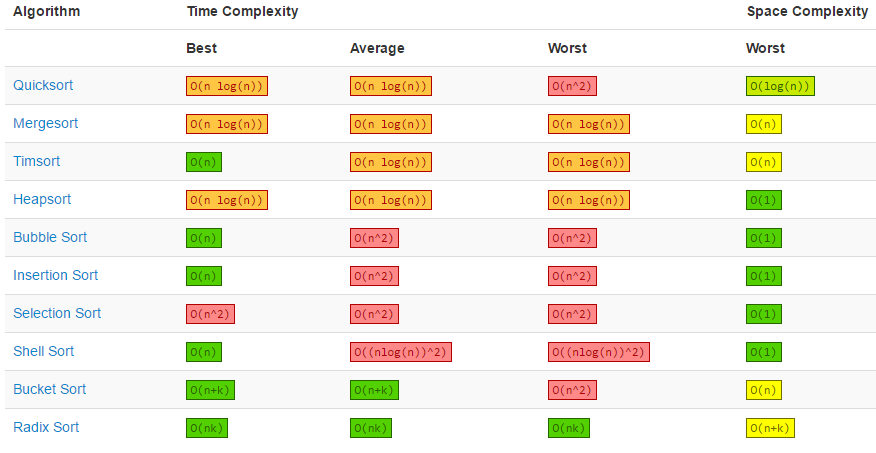
어려웠던 점.

과제 중 어려웠던 점은 과제에 대한 설명이 너무 부족했다는 것이다. Algorithm이나 data structure을 어렵지 않을 것을 사용하여 난이도는 없었지만 Project PDF에 말고 Facebook에 올라오는 새로운 정보들 때문에 계속 코드를 바꿔야 했다.

다른 어려운 점은 코딩을 너무 오랜만에 했고 python으로는 처음 개발해봐서 생소한 것 많았다는 것이다.

미래에 더 시간이 있다면.

시간이 많이 있었다면 TimSort를 구현해 보고 싶다. Timsort는 여러가지 알고리즘을 짬뽕시켜 만든 adaptive 알고리즘이다.



Time complexity를 보면 알겟지만 Timsort가 가장 뛰어나다. 구현은 해보았지만 MinRun이라는 timsort를 상황에 따라 adapt 하게 해주는 개념을 자세히 이해 못해서 그냥 포기하였다.