Data Structure Report

Sunghwan Hong 2015410121

1. I used RB Tree to efficiently store the IDs of Users. As I used RB Tree as my main data structure to traverse nodes, the time complexity greatly reduced than other data structure. Making RB Tree was tricky but it helped me to expand the variety of data structure I could use. Using RB Tree, more specifically ID in each single node, I utilized them to make another ‘node’. This ‘node’ is made to store information of users, tweets and others. Then as members of that ‘node’ I used LinkedList to show the relationship of users. Who follows who and who is followed by whom. Using linked list allowed me to store follow users and it allowed me to save memory. I also used Queue, but as RB Tree needs Priority queue, I had to import Heap. To heapify and sort the data in favor of RBTree’s taste, and also to greatly reduce the amount of time taken to sort all of those without Priority Queue and Heap, I utilized them. Furthermore, there was an instruction that asked me to find top 5 most tweeted users. This needed sorting in order to find the top 5. So just simply inputting in priority queue, I could find top 5. Lastly, I used Hash in order to not let collision between words when inserting word into tree. If the collision happens it would make my program’s quality low. And Hash’s time complexity is O(1) so if I use Hash for applying RB Tree, I expected to greatly reduce calculation. For question 2, to find top 5 most tweeted words, I applied hash table in order to find the tweet number directly.
2. My expected performance is as I could not do number 8 and 9 , which is SCC and shortest path, I expect other options work perfectly. And time complexity must be logx (rb tree) to search, delete, insert, and multiply to time complexity of linked list, n and I’m not quite sure but it must be not that huge that could result in malfunction of my program. Update : I calculated the time complexity. So creating RB Tree is log n. then id is stored in nodes of RBTree and linked list is connected along it. So searching for specific user’s information would take nlogn. Deleting would also take nlogn. When getting top 5 results, as I used heap sort, it should take nlogn as well.
3. My expected performance was to perfectly do every number other than 8 and 9. However I was able to do perfectly for number 6 and 7 but while I was fixing bugs, I accidentally fixed something wrong and number 6 and 7 doesn’t seem to work. I should be trying to fix it but as I tried for more than a full day to just to fix those number 6 and 7 , and I still couldn’t fix it, I hope to fix it in the future. UPDATE: I Fixed number 6 and 7 by finding unnecessary statement and getting rid of it. I was ‘free’-ing something that shouldn’t be freed. I eliminated it from my code, and number 6 and 7 works perfectly. 2016/6/24 21:57 2nd UPDATE : I found out that option 0, read the data file, should be showing the results only(?). Not reading the txt files. Txt files should be read when the program starts. However my program reads the txt when 0 was inputted. So in the future I may change the function of option 0. If I input 0 twice, it makes an error. I asked professor about this issue, and he told me that this isn’t good but still if other options all work, it would be fine. However in the future, I could apply to make option 0 to show current statistic. 2016/6/25 21:35