Brandon Barrett
5 November 2019

Project 2 Final Submission

1. **Problems Encountered in Implementation**

- I really didn't have too many problems implementing the queries in the second half of the project.
- The largest single obstacle was creating a variant of building a max heap to keep track of both the storm_event nodes being swapped and the number fatality_events associated with them.
    - The solution to this was to create an integer array that held the number of fatality_events associated with each node and swap them when the storm_event nodes were also swapped in the heap.
- For the binary search tree the hardest aspect was calculating the height of the BST subtrees.
    - For this I turned to outside sources for some insight into the best way to traverse the whole tree to calculate the height.

2. **Known Bugs or Incomplete Implementation**

- Fixed a problem with the load factor not calculating the largest chain correctly, fixed
- To the best of my knowledge all parts of the project were implemented successfully.
- I have not run across any bugs or sudden segmentation faults associated with the project while testing the submitted code.

3. **Significant Interaction with Peers**

- Tingyu Luo – Mostly discussed the BST query
    - I helped her with creating the BST since it seems like she was attempting to create a heap for that query as well.
    - We also caught some of each others segmentation faults from creating new dynamically allocated structures

4. **Citations to External Books or References**

- Introduction to Algorithms textbook – Used to assist with creating the max heap
    - Cormen, Thomas H.., et al. Introduction to Algorithms. Massachusetts Institute of Technology, 2009.
- GeeksForGeeks.com – Used it to help me calculate the height of the BST left and right subtrees
    - Multiple Authors. "Write a Program to find the Maximum Depth or Height of a Tree." *GeeksforGeeks*, 23 July 2019, https://www.geeksforgeeks.org/write-a-c-program-to-find-the-maximum-depth-or-height-of-a-tree/

Project 2 Milestone

5. **Problems Encountered in Implementation**
    - The largest problem I had was simply implementing the structs at the beginning. My solution was to declare the annual storms and storm events statically, but to implement the linked lists and hash table dynamically with pointers.
        - The annual_storms and storm_events were static because it was possible to determine how much memory they would take up at run time and using excessive pointers here would be wasteful.
        - The hash_table and all the linked lists were made dynamically with pointers because it wasn't possible to determine their size or how full they would be initially which made using NULL as an indication of emptiness very useful. Also, pointers to null cut down on wasted memory from allocated, but completely unused structs.
    - Another problem encountered was reading the CSV files initially and I had to quickly brush up on my stringstream and conversion functions in order to get past this immediate hurdle of simply reading the file data in.
6. **Known Bugs or Incomplete Implementation**
    - At the present time there are no known bugs
    - The only incomplete implementation is how to handle empty fields. They are currently handled, but they may not be up exactly what the project specifications ask for.
7. **Significant Interaction with Peers**
    - Tingyu Luo – Mostly discussed importing data from the CSV files
        - In the end we went about importing the information in two significantly different ways, but to the same result
        - I also helped her with some issues in dynamically allocating the structs
    - Asked a couple of questions on Piazza to clarify some of the output implementation
8. **Citations to External Books or References**
    - StackOverflow.com – Helped me out a lot with figuring out a simple way to simply use the commas in CSV files as delimiters
        - User2249683. "C++ Reading CSV File." *Stack Overflow* 12 November 2013, 18:02, stackoverflow.com/questions/19936483/c-reading-csv-file/19936571#19936571
    - Cplusplus.com – Used primarily for specific cases where I got stuck on implementation and needed exact functional definitions
    - C++ Cookbook from O'Reilly – Used it for simplifying some of my string-based functions
        - Stephens, Ryan, et al. "Chapter 4: Strings and Text." C++ Cookbook, O'Reilly, 2006, pp. 139–196
    - Introduction to Algorithms textbook – Used it to assist with implementation of the hash table and calculating load factor
        - Cormen, Thomas H.., et al. Introduction to Algorithms. Massachusetts Institute of Technology, 2009.
    - GeeksForGeeks.com – Used it to help me to further visualize hashing with separate chaining when I got stuck
        - Shubham, Rana. "C++ Program for Hashing with Chaining." *GeeksforGeeks*, 12 February 2018, https://www.geeksforgeeks.org/c-program-hashing-chaining/.