Edit

Assignment 2 Group 25 Discussion

18

Vandit Jyotindra Gajjar

All sections

Hello everyone,

This is the discussion thread for the group 25 that includes ['a1749431', 'a1779153', 'a1786974', 'a1792052'].

I am Vandit Gajjar[a1779153]. I will update the reviews soon on this thread for other students and you're also welcome to discuss this first. Thank you.

Search entries or author | Unread | Subscribed

Write a reply...

Vandit Jyotindra Gajjar

(https://myuni.adelaide.edu.au/courses/54378/users/109438)

4 May 2020

You can find my code in this attachment.

Cheers!

main.cpp (https://myuni.adelaide.edu.au/files/6440746/download?download_frd=1&verifier=XAgbVZ3TXDyyiugwlaMR8WeXGlOehyxLo2AL1hSv)

Reply

Li-Cheng Chen

(https://myuni.adelaide.edu.au/courses/54378/users/107726)

4 May 2020

Hi,

I am Leo[a1786974], the attachment is my code.

Cheers!

main.cpp (https://myuni.adelaide.edu.au/files/6440791/download?download_frd=1&verifier=ckqSVGx49PLwx6a33azTRnxUltXcRgQSLwWaMvsy)

Reply (1 likes)

Ka Yiu Eric Ma

(https://myuni.adelaide.edu.au/courses/54378/users/115491)

4 May 2020

Hi all,

Attached please find my code.

Eric

main.cpp (https://myuni.adelaide.edu.au/files/6442386/download?download_frd=1&verifier=Z7TQd5FQNQws7F6SWgt85cHMjU4hahLvvVOt89Df)

Reply (1 likes)

Ka Yiu Eric Ma

(https://myuni.adelaide.edu.au/courses/54378/users/115491)

4 May 2020

My UID is a1792052

Reply

Jasper Winter

(https://myuni.adelaide.edu.au/courses/54378/users/81024)

Tuesday

Hi guys! Jasper here, here is my assignment

main.cpp (https://myuni.adelaide.edu.au/files/6451892/download?download_frd=1&verifier=VfYASduZAqXX8FCzsl7RSs1BA6QP3dov969ihJSl)

Reply (1 likes)

Jasper Winter

(https://myuni.adelaide.edu.au/courses/54378/users/81024)

Tuesday

my ID is [a1749431]

Reply

Ka Yiu Eric Ma

(https://myuni.adelaide.edu.au/courses/54378/users/115491)

Wednesday

To a1779153:

- I did not know that vector of input data can be initialised easily using the istream_iterator of istringstream and a default istream_iterator. Lesson learned! :)

- To improve readability:
  - For the function treeModifier, it's better to add a comment to indicate that the modification is insertion when removeOrAdd is 1 and the modification is deletion when removeOrAdd is 0.
  - Add a comment to briefly explain what the variable "difference" means in the struct of treeNodeObject
  - Define global constant variables "LEFT" and "RIGHT" as 0 and 1 respectively, and use them as index of child array of a treeNodeObject, so to clearly state whether you are referencing to the left/right child of a parent node.
  - Add comment to explain what the variables "leftSidetreeModifier" and "tmpDeltaSignature" mean. It seems the variable "leftSidetreeModifier" attempts to modify left subtree when it is equal to zero. If that is the case, it could be confusing as conventionally zero == false.
- It is interesting that AVL tree can be implemented using only one pair of rotation operations. Thanks for letting me know there is one more way to implement the AVL tree

Reply (2 likes)

Ka Yiu Eric Ma

(https://myuni.adelaide.edu.au/courses/54378/users/115491)

Wednesday

To a1786974:

- I did not know that in C++, default and parametric constructors of struct can be defined in this way. It can save amount of code for creating new node during insertion with such parametric constructor. Lesson learned! :)
- Is there any reason why function name "PRE", "POST" and "IN" are capitalised? To my knowledge, capitalised naming is for global constant variables. It would be great if you can share why if it follows another convention.
- I did not know that tree rebalancing can be achieved by only one pair of rotation operations when the rotation operation is defined in this way and are performed in four different ways under four different conditions of balance values of current and inner child node. Lesson learned! :)
- To improve readability:
  - function name "PRE", "POST" and "IN" could be renamed as "print_pre", "print_post" and "print_in" to better indicate them as functions for printing nodes

- To improve code maintainability by reducing duplicated coding logic:
  - As the function left_rotation and right_rotation follow the same logic for mirrored nodes, they can be combined into one function with an additional parameter “isLeft” as a flag to indicate whether the rotation is left or right.

Reply (2 likes)

Ka Yiu Eric Ma

(https://myuni.adelaide.edu.au/courses/54378/users/115491)

Wednesday

To a1749431:

- It is better to add comment to describe “str[i] - 48” is for converting str[i] from string to integer as it is not obvious.
- It would be more simple to get the value of each move by first getting substr(1) of each input string, then using the function stoi() to convert it into integer
- For functions like upHeight() and upBalance(), they are clearly operations for internal use. You may consider to defline them as private, rather than public methods.
- When referencing the private attribute (e.g. root), you can improve the code readability by adding “this->” to clearly state that those variables are private attributes.
- function name “pre”, “post” and “inorder” could be renamed as “print_pre”, “print_post” and “print_inorder” to better indicate them as functions for printing nodes
- As the functions “LRotation” and “RRotation”, “RLRotation” and “LRRotation” follow the same logic operating on mirrored nodes, they can be combined into one function with an additional parameter “isLeft” as a flag to indicate whether the rotation is left or right.

Reply (2 likes)

Vandit Jyotindra Gajjar
(https://myuni.adelaide.edu.au/courses/54378/users/109438)

Thursday

Review for a1786974:

Hi, a1786974,

Summary:

- First of all many congratulations on your submission of the code. All test cases have been passed easily.

Strengths:

- The code is written in a very effective manner, it's easy to follow as you have added necessary comments on a certain logic. I want to learn this habit of commenting in my code where it is necessary.
- Overall, it is a very good code, I will mark the perfect score for this submission even though there are some minor issues.

Issues:

- I believe that one should add references at the bottom, although I also support that you did was good where you have added source/reference from what you learned from it, you added at certain logic portion. I'll suggest that you should something similar like cross-referencing where you add a reference at the bottom but you cite at the certain logic portion, so if the reviewer wanted to see that it'll be easy to follow up.
- Also, spacing is another minor issue that I found what we both need to follow that one should properly give space when logic or function implemented, it'll eventually increase readability.
- I'm just curious can you able to implement Switch-case style manner in the function of get_balance, I do not know, just posing that it might work or work not, and how? Just Curious! :)

Encouragement:

- As a beginner in programming, I was able to get your logic, so I appreciate your efforts as it is really easy to follow your intuition behind the development.

Final Comments:

- Perfect code, easy to follow, proper commenting, proper indentation (Which I will learn from you, and going to improve on my next assignments). Some minor issue mentioned in the issues section, which I believe one should follow it, for readability purpose. Overall I'll give full marks.

I sincerely apologize for my late review of your submission.

Reply (1 likes)

Vandit Jyotindra Gajjar
(https://myuni.adelaide.edu.au/courses/54378/users/109438)
Thursday

Review for a1749431:

Hi, a1749431,

Summary:

- First of all congratulations on your submission. Even though not all test cases have been passed, but at least you have given your best which I appreciate a lot.

Strengths:

- Mostly the code follows if-else conditions, which is good as it seems that you're trying to code from the bottleneck.

Issues:

- I think one should try to provide spacing between functions which is a minor issue, but it'll be good for readability.
- I also believe that certain naming of variables and functions are inappropriate, which you should try to consider in your next assignment. For example Tree T >> Tree tree, sInput >> stringInput, LRRotation >> leftRightRotation, RLRotation >> rightLeftRotation. [1] is an excellent resource for you to find out about naming conventions.
- In your case, LRRotation-RLRotation and LRotation-RRotation seem to be working similarly except if condition and in pointers, so maybe it'll be good to create a function which should incorporate all of this thing and reduce your code's

lengthiness. Just a suggestion.

Final Comments:

- Good comments on certain logic, indentation is also good, Certain cases haven't been passed but what I believe is I truly appreciate your efforts to reaching to a solution from zero to getting some decent score. Just a quick suggestion that have you tried to print your data and try to check line-by-line where might be the error? I'll try to come up with an answer in 2-3 days max. Cheers!

I sincerely apologize for my late review of your submission.

References:

[1] https://google.github.io/styleguide/cppguide.html (https://google.github.io/styleguide/cppguide.html)

Reply (1 likes)

Vandit Jyotindra Gajjar
(https://myuni.adelaide.edu.au/courses/54378/users/109438)
Thursday

Review for a1792052:

Hi, a1792052,

First of all congratulations on your submission of the code. All test cases have been passed successfully.

Strengths:

- Good commenting, indentation, the naming of functions, and logic. This is a far better good submission than mines. So many good things to learn from your overall code, again thank you for this.
- Testing what you have done to print out the data is good, which will help to debug the code in case of error occurs, so this will be helpful.

Issues

- I was unable to find issues as most of the things I encounter useful for me in future assignments.
- I believe that if you could have added references regarding where you have learned from the logic except lecture slides this will encourage us to get an idea about the development process.

Final Comments:
-- Easy to follow code, every function has a description which is a good thing as documentation wise it seems professionally, I came across some good things which I should take care of in the future.

I sincerely apologize for my late review of your submission.

Reply (1 likes)

Li-Cheng Chen
(https://myuni.adelaide.edu.au/courses/54378/users/107726)
Thursday

a1779153: I have learned the usage of node in your treeRotation and treeModifier function, which inspires me to try implementing the assignment in a different data structure! I really appreciate the comments, label and reference in your code. They are really well described and implicit for understanding. Minor issue will be the indentation for ease to read, but it will not affect the logic implemented in the function. Thanks for your comments, I will try to figure out whether it is possible to use switch-case style manner in my get_balance function.

a1749431: Nice implementation! The comment and indentation are clear and easy to read. If you could use some print functions to trace line-by-line code, it may help you to figure out the error. The rotation part maybe one of the possible errors if the determination function is wrong, as well as the deletion function.

a1792052: I have learned a lot from your code style and logic in this assignment. The code is in well indentation, comment, naming and easy to follow. The description is detailed and straightforward, however, if you can provide some reference about how you develop, it would be really appreciated. Thanks for your comments, especially pointing out the naming issue in the function. Simply I just follow the PRE/POST/IN action mentioned in the guide of the assignment, naming each move in the exact same name. I will fix it into print_ for better understanding and combine the same logic with the function left_rotation and right_rotation.

I sincerely apologize for my late review of the submission.

Reply (1 likes)

Jasper Winter
(https://myuni.adelaide.edu.au/courses/54378/users/81024)
Thursday

Hi guys, Jasper here. I appreciate your patience in waiting for my review. Thanks for all your reviews!

a1792052

Style:

The styling in your code is good. The commenting is thoughtful and used sparingly.

Variable and function naming is also easy to understand.

Code:

I like how conditional operators were used when defining the node pointers in your rotate function. This allows you to easily use the same code for varying operations; I think I will try to implement something similar.

I may suggest using one balancing function for both deletion and insertion that iterates through the tree when an operation is performed, however this can be implemented in different ways depending on whether large trees are expected (meaning run time compounds).

a1779153

Style:

Good helpful commenting and only where it makes sense. Variable and function naming is good and together with the large indents and comments makes for a very easy to read code. Good job!

In my text editor (Atom) the indents were a little weird and it is very possible that this was a compatibility issue, however I will treat it as if it is how it was meant to be:

Indents seem to be a little excessive in areas where it makes determining what scope certain code belongs to difficult. See the attached image to see what I mean (once again it may just be a software issue).

Code:

Interesting to see the lack of parent pointers in the struct definition. I assumed this was almost necessary in performing the operations required but not only is it clear that it isn't, your code is far more efficient than mine. It is very compact and efficient with no clearly unnecessary lines or code.

Overall great code and no real improvements needed as far as I can see! Great work.

a1786974

Style:

Great commenting and variable/function naming. Comments are good however a little excessive at times. For example in your printing functions it is already clear from the code that when the tree is empty to print empty, this is not needed to be commented in.

Code:

Overall very compact and code. Functions are efficiently implemented in a very space efficient manor. Once again I am impressed that the function runs far better without the implementation of a parent pointer (this was actually something that made my code far more complex than it may have needed to be).

One thing I would recommend is to not use a struct for your tree implementation. Structs should usually be used for small instances that usually represent a single value (int, bool, etc). For more information on why I would suggest reading: **Choosing Between Class and Struct (https://docs.microsoft.com/en-us/dotnet/standard/design-guidelines/choosing-between-class-and-struct?redirectedfrom=MSDN)**

You may have a good reason for using a structure here however, so this is merely a suggestion!

Other than that your code is great!

code.png (https://myuni.adelaide.edu.au/files/6468279/download?download_frd=1&verifier=jV6oH1xaaklUAHxM7Er3zLgEaFelublWtXEfYkSM)

Reply (1 likes)

Ka Yiu Eric Ma
(https://myuni.adelaide.edu.au/courses/54378/users/115491)
Saturday

Reflection:

Thanks everyone for taking your time to review my code! Your positive feedback encourages me to continue with the following good practice of coding:

- Using variable and function name that can be easily understood by other programmers

- Writing clear and good comments to improve coding readability

I have learnt that:

- The method introduced in the lecture is not the only implementation of AVL tree. Using another method, it is possible to implement the AVL tree using only one balancing function with only one set of rotations for balancing the tree following insertion and deletion.

I should improve my future work by adding references:

- It is indeed a good suggestion. Even I only referred to the lecture materials without googling further to work for this assignment, it is a good practice to explicitly mention what lecture materials my implementation is based on. If in the future I do not remember the node rotation processes I implemented, I can easily locate the related lecture materials if I keep this record clear.

In reply to the suggestion of using same rebalancing function following insertion and deletion:

- I did not do that as according to the lecture materials, the type of rotation performed following insertion is based on whether the inserted node is at the left-left/left-right/right-left/right-right of the first problematic node. However, the type of rotation performed following deletion is not based on the same logic: It is based on whether the height of outer subtree is greater than or equal to the height of inner subtree at another side of the deleted node. I would be grateful if you could share how these two logic could be elegantly combine into one function for me to improve my coding skills further.

Cheers,

Eric

Edited by **Ka Yiu Eric Ma (https://myuni.adelaide.edu.au/courses/54378/users/115491)** on 9 May at 16:57

Reply (1 likes)

Li-Cheng Chen

(https://myuni.adelaide.edu.au/courses/54378/users/107726)

Saturday

Reflection:

Thanks everyone for taking your precious time to review my code! I have made the following change of my code based on the review:

- Using variable and function name that can be easily understood and fulfilled with the convention (function name “PRE”, “POST” and “IN” change to “print_pre”, “print_post” and “print_in”)

- Improve code maintainability (left_rotation and right_rotation coding logic

- Reference added at the bottom

- Code format and style (spacing and indentation modified by software provided by the professor)

What I have learned:

- Different logic in implementing AVL tree

- Practice reading code and providing useful comments

Finally, really appreciate that our group provides lots of useful suggestions for my code and I will keep learning!

Cheers

Leo

Reply (1 likes)

Jasper Winter

(https://myuni.adelaide.edu.au/courses/54378/users/81024)

Yesterday

Hi all and thanks for your reviews. While I have not managed to pinpoint the problem in my code causing slight variations in *some* inputs here is what I have taken away from your feedback:

- My largest weakness I believe to be my naming conventions (something that was remarked upon a lot). From now on I will make sure that all functions and variables are far more descriptive as to their functions and implementations in the code.
- From reviewing your code and in the feedback received I decided to merge some of the functions that use very similar code structure into one and use conditional operators to correct the slight variations in implementation.
- I had implemented code using string vectors and atoi operations to parse the inputs but reverted during debugging (this will be restored since its far more efficient).

A quick note on my debugging operations and what I used to find problems. I have used <cassert> from the std library to check for errors in code; importing code to visual studio to debug with code breaks; cout lines with exact values at chokepoints that could be causing issues:

e.g. cout << "Left Right Rotation of: " << node->key << " with: " << nodeLR->key << endl;

This was incredibly helpful in finding the cause to segmentation faults however wasn't able to highlight the slight output difference.

Here is a sample:

Output received 28 16 11 5 12 18 17 21 19 27 70 53 42 37 34 39 51 57 54 61 58 84 79 73 72 77 82 83 95 93 88 94 99

Output should be 28 18 12 11 5 16 17 21 19 27 70 53 42 37 34 39 51 57 54 61 58 84 79 73 72 77 82 83 95 93 88 94 99

Hopefully with a little bit more work I can find the issues.

Thanks again for your help.

Reply (1 likes)

Vandit Jyotindra Gajjar

(https://myuni.adelaide.edu.au/courses/54378/users/109438)

0:15

Dear Team Members,

I would like to take this opportunity and want to say thank you very much for your feedback.

As most of you have noted that I have tried to work out something different in my code and which is right. I tried simple logic at the first by following the lecture slide, then after looking carefully at my code, I wanted to further optimize it in just 1-2 functions, thus after working tirelessly by trial and error, I have updated my code.

Also, all of you have noted that my indentation is not proper and for which I'm thankful as the actual reason is as follows. I'm using Notepad ++ for almost all of my assignments in this and the other subjects as well. The idea behind using that is it gives you the experience of white-board coding like interview style. As many of you know that while giving technical interviews, we have to write our code or logic in whiteboard or paper. Thus using Notepad ++, it gives you the same feeling. So the only debugging tool is that you have to solve the problem using a line-to-line manner, thus it gives good experience the same as technical interviews and while Notepad++ has some issues with meditation, from the next assignments, I'll take for those mistakes next time.

Once again, I'm thankful to all of you as I have learned some of the mistakes and also got a good idea from your codebase on how to workout with other issues.

Cheers!

P.S: Sorry for my late reflection, as I do not have an internet connection in my home right now, so I have to upload this from the university. Hope you understand.

Reply