

Creativity Challenge – Code Review Template

Coder Name: Nolan Downey

Reviewers Names: Connor Ruff, Kelly Buchanan

Creativity Challenge Number: 4

Part 1 - Correctness and Defects – 150 points towards Creativity Challenge Grade – Good faith documentation of these approach will receive full credit.

Does the program do what you expect?

- If the code compiles and produces the correct output, to earn the 150 points:
- Describe in detail why your reviewers believe it works
 - Describe their assessment of your proposed solution based on problem statement and the result generated just for correctness (efficiency and memory are in Parts 2 and 3)
 - Show the code run for several test cases
- If the code does not compile or produce the correct output, to earn the 150 points:
 - Describe the currently existing compiler/Makefile linker errors, if any exist.
 - Describe the code segments that are potentially contributing to these errors.
 - Describe the currently incorrect outputs and logical defects, if any exist
 - Describe the code segments that are potentially contributing to these errors.

Reviewer Comments:

My team analyzed my code and concluded that it compiles and produces the correct output. Using test cases from the given video and from our own testing, my team and I concluded that my code runs with no errors and produces correct outputs. Below, you can see one example of the testing of my code to find the lowest common ancestor of two given inputs.

```
ndowney@student12:~/DSInClass/CC4$ make CC4
g++ -m64 -std=c++11 -weffc++ -O0 -g -Wall -Wextra -Wconversion -Wshadow -peda
ntic -Werror -lm -c ~/DSInClass/CC4/programs/CC4.cpp -o /escnfs/home/ndowney
/DSInClass/CC4/objects/CC4.o
g++ -m64 -std=c++11 -weffc++ -O0 -g -Wall -Wextra -Wconversion -Wshadow -peda
ntic -Werror -lm -o ~/DSInClass/CC4/executables/CC4 ~/DSInClass/CC4/objects/
CC4.o
~/DSInClass/CC4/executables/./CC4
-----
First, enter the number of elements in the tree: 9
Next, insert 9 into the tree: 8 3 10 1 6 14 4 7 13
The initial tree is:
Tree - In-Order Traversal: 1 3 4 6 7 8 10 13 14
Tree - Pre-Order Traversal: 8 3 1 6 4 7 10 14 13
Tree - Post-Order Traversal: 1 4 7 6 3 13 14 10 8
Insert a pair to check - [First] [Second]: 1 7
Find LCA for 1 and 7...
3
Continue? y/n: y
Insert a pair to check - [First] [Second]: 4 7
Find LCA for 4 and 7...
6
Continue? y/n: y
Insert a pair to check - [First] [Second]: 6 6
Find LCA for 6 and 6...
6
Continue? y/n: y
Insert a pair to check - [First] [Second]: 8 13
Find LCA for 8 and 13...
8
Continue? y/n: y
Insert a pair to check - [First] [Second]: 13 14
Find LCA for 13 and 14...
14
Continue? y/n: y
Insert a pair to check - [First] [Second]: 14 13
Find LCA for 14 and 13...
14
Continue? y/n: y
Insert a pair to check - [First] [Second]: 4 13
Find LCA for 4 and 13...
8
Continue? y/n: n
..Exiting Program...
ndowney@student12:~/DSInClass/CC4$
```

Part 2 – Performance – 125 points towards Creativity Challenge Grade – Good faith documentation of these approach will receive full credit.

The team's assessment of the run-time efficiency of your proposed solution.

- Does this program use the best Data Structure for runtime given the problem constraints?
 - If your teams believe so, state why.
 - If your team believes not, state why not.
- Does this program run in the most efficient manner possible given the available resources? (i.e., not using unnecessary loops and complex approaches when a Data Structure could simplify the approach)
 - If your teams believe so, state why.
 - If your team believes not, state why not.

Reviewer Comments:

My team analyzed the run-time efficiency of my proposed solution and determined that it is a very efficient way to find the LCA of two given inputs. We thought that the use of another data structure would increase the run-time of this code, and for these reasons we thought recursion was the best solution to this problem. I initially analyzed each of the different possibilities of approaches for this problem, and also came to the conclusion that recursion eliminated any unnecessary loops that would increase run-time. My team and I both believe, at this point, that a Data Structure would only complicate the approach more than it already is.

Part 3 – Memory – 125 points towards Creativity Challenge Grade – Good faith documentation of these approach will receive full credit.

The team's assessment of the space efficiency of your proposed solution.

- Does this program use the best Data Structure for memory efficiency given the problem constraints?
 - If your teams believe so, state why.
 - If your team believes not, state why not.
- Does the program make effective use of the heap (i.e., not unnecessarily storing redundant copies of information to produce the solution when a better approach exists), pass/call by reference, function, and method calls?
 - If your teams believes so, state why.
 - If your team believes not, state why not.

Reviewer Comments:

My team pointed out that I did not include any data structure other than the Binary Search Tree for this problem. We were given the hint that maybe a dynamic array or a singly-linked list could be involved, but we believed this would increase memory allocation and decrease run-time efficiency, so I avoided the use of extra data structures. I believed, and my team confirmed, that recursion would make the most effective use of the heap, and writing the code as a method built into the BST would also be the most beneficial approach.

Part 4 – Finding Better Solutions – Optional

Let's say that you have reviewed your group mates code and determine there is a better approach than the one you are taking. A concern you might be having right now would be "if I change my answer, is that Academic Dishonesty?" Here is how you can modify your solution post Code Review and maintain academic integrity:

Describe the changes - State the change in "Correctness and Defects", "Performance", and/or "Memory", as appropriate, that contributed to your solution, and how you wrote your own version of the changed code to meet these changes.

Cite: Name your team member or classmate who gave you the idea, to give appropriate attribution.

Coder Comments:

It seems as though my team and I all had similar approaches to the problem. We all thought recursion was the best for memory allocation and run-time efficiency, but all of our methods were a little bit different while using the same logic. At this time, I do not know if there is any better solution.