

Task 5 Answers for DSW

Unlike the traditional implementation, we limit the number of total rotations to create a partially left skewed vine instead of creating a completely right skewed vine. Additionally, phase 2 operates in a way that makes a tree that is not completely balanced as opposed to a completely balanced tree. For phase 1, since we are skipping some rotations for subtrees that are smaller than 3, we are reducing the number of rotations which leads to better efficiency despite the unbalanced result. In a similar fashion, phase does something similar. Instead of perfect balance, we are going for a height that is somewhere around $2\log_2(N)$. This means that the tree is not perfectly balanced unlike the original implementation. For this implementation, we are trading off better efficiency for a less perfectly balanced tree. For searching, the implementation would require slightly more time, but not enough to make a huge impact on the overall complexity of the function. For another variation, we could do something like the semi-splay tree where we limit the number of rotations that are performed by some arbitrary number we decide. It would lead to a similar situation to the current one where there is still a general skew to one side or the other. This would lead to a better time complexity whenever we need to rebuild the tree, but it could also lead to issues where the tree is very unbalanced based on how much data we have and what we set the limit to. Overall, it would present an option that, if implemented well, could lead to better efficiency and better predictions of what our worst-case scenario would only lose value to the balancing limitations.