

Implementation:

In order to check whether the brackets of the entered string are balanced or not, a stack was needed to store the brackets. When an opening bracket was detected in the string, it was immediately pushed to the bracket stack and the program continued to check the rest of the string. When a closing bracket was detected, it popped the most recently used opening bracket and if the types matched (ex. '(' with ')', '{' with '}', '[' with ']'), then the Boolean value which checked balance, would stay 'true.' If the detected bracket does not match the popped bracket, then the Boolean value would change to 'false'. Finally, if there are any remaining brackets in the stack left after the entire string was checked, then the Boolean value would be 'false' because no corresponding bracket would be present to close them.

Time Complexity:

Best Case: $O(1)$

Average Case: $O(N)$

Worst Case: $O(N)$

Memory Complexity:

Best Case: $O(1)$

Average Case: $O(N)$

Worse Case: $O(N)$