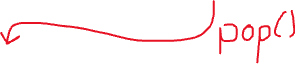


We can see it as a recursive problem. Each outer bracket that has nested bracket inside is a parent node. We can build a tree for each input.

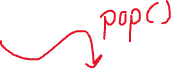
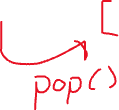
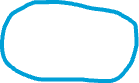


Stack every character until the first closing bracket ] and do not stack it but go backward and pop characters to store the current substring belonging to the closing bracket.



Once we reach the first opening bracket [ , we pop it but we don’t store it.

We continue to go backward everytime we detect a digit from 0 to 9 we store it in a string variable for the number to multiply until we reach an element that is not a digit.



We build the substring then we stack it inside the stack.



We continue iterating through the string and stacking



Each iteration of outer loop does not trigger a full iteration of the inner loop.

