

An **Efficient Solution** can solve this problem in $O(n)$ time. The idea is to store indexes of previous starting brackets in a stack. The first element of the stack is a special element that provides index before the beginning of valid substring (base for next valid string).

1) Create an empty stack and push -1 to it.

The first element of the stack is used
to provide a base for the next valid string.

2) Initialize result as 0.

3) If the character is '(' i.e. `str[i] == '('`),
push index 'i' to the stack.

2) Else (if the character is ')')

- a) Pop an item from the stack (Most of the
time an opening bracket)
- b) If the stack is not empty, then find the
length of current valid substring by taking
the difference between the current index and
top of the stack. If current length is more
than the result, then update the result.
- c) If the stack is empty, push the current index
as a base for the next valid substring.

3) Return result.