Алгоритмы и структуры данных поиска

ДЗ №1, задача 1-2. Теоретическая часть.

*Описание алгоритма.*

Рассмотрим первую правую скобку в последовательности. Пусть эта скобка встречается в позиции t > 0. (Она не может стоять в начале, т.к. ее ничто не сможет закрыть). Тогда скобка в позиции t – 1 является левой скобкой и парой к скобке на позиции t. Действительно, если бы какая-то левая скобка на позиции l < t t − 1 была парой к правой скобке на позиции t, то пара к левой скобке на позиции t − 1 находилась бы правее t, тогда эти две пары пересекаются, что противоречит определению. С другой стороны, если самая первая правая скобка стоит на позиции t > 0, то строку еще можно продолжить так, чтобы она стала корректной: нужно дописать правых скобок столько, чтобы закрыть все левые скобки, и остановиться. Поэтому самая первая правая скобка и левая скобка сразу перед ней соответствуют друг другу. Теперь, если выкинуть эту пару скобок из строки, корректность оставшейся строки определяется тем же самым способом, так как пара скобок, стоящая непосредственно рядом, без промежуточных скобок, не мешает выполнению ни одного из наложенных условий. Из этих соображений вытекает следующий алгоритм решения задачи.

Создадим в самом начале пустой стек. Далее идем по строке слева направо. И в зависимости от прочитанного символа и состояния стека делаем следующие действия:

Если символ – открывающая скобка, то просто кладем его в стек и переходим на следующий шаг.

Если символ – закрывающая скобка, то если стек пуст или на вершине стека лежит символ не составляющий пару текущему, то на данном шаге последовательность уже не правильная, при этом ее можно было дополнить до правильной последовательности до этого шага. То есть нужно вывести длину прочитанного префикса уменьшенного на один. Если на вершине лежит парная скобка к текущей, то удалим ее их стека и двигаем дальше.

После того как строка прочитана выводим CORRECT, если стек пуст. И длину всей последовательности иначе.

*Оценка времени и памяти.*

При обработке каждого символа требуется константное время. Алгоритм проходит по строке константное число раз, значит временная сложность O(n). Памяти требуется также O(n) на хранение строки и стека, количество элементов в нем никогда не превосходит n.