* [Правильные скобочные последовательности. Проверка на правильность.](https://foxford.ru/lessons/32269/conspects/1)
* [Правильные скобочные последовательности. Подсчет количества](https://foxford.ru/lessons/32269/conspects/2)

**Правильные скобочные последовательности. Проверка на правильность.**

**Определение правильной скобочной последовательности**

Одним из применений стека является алгоритм проверки правильности скобочной последовательности. Рассмотрим какое-либо выражение, в котором могут встречаться скобки трех видов — круглые «(» и «)», квадратные «[» и «]» и фигурные «{» и «}». Теперь уберем все символы, кроме скобок. Полученную последовательность будем называть «правильной скобочной последовательностью».

Например, последовательность «([])» будет правильной, а последовательность «([)]» - нет (нарушен порядок вложенности скобок.

Более формальное определение правильной скобочной последовательности строится индуктивно. Прежде всего, пустая строка — это правильная скобочная последовательность. Если A — правильная скобочная последовательность, то «(A)», «[A]» и «{A}» также являются правильными скобочными последовательностями. Наконец, если A и B — правильные скобочные последовательности, то их конкатенация AB также будет правильной скобочной последовательностью.

**Проверка скобочной последовательности на правильность**

Это определение позволяет сформулировать алгоритм проверки скобочной последовательности на правильность. Найдем в последовательности пару подряд идущих парных скобок — например, «()». Удалим эту пару из последовательности. Будем продолжать так пока есть такая пара. Если в результата получилась пустая строка, то исходная последовательность была правильной.

Но поскольку удаление подстроки из середины строки выполняется за линейное время, то такой алгоритм будет иметь сложность O(n2).

Использование стека позволяет улучшить этот алгоритм — встреченная закрывающая скобка должна быть парной к последней встреченной открывающей, после чего их необходимо удалить. Для хранения открывающих скобок будем использовать стек. Пройдем по строке от начала до конца. Если очередной символ — открывающая скобка, то добавляем ее в стек. Если закрывающая — то проверяем, что стек не пуст и скобка на вершине стека парная к данной закрывающей. При соблюдении условий открывающая скобка удаляется из стека, иначе алгоритм заканчивает свою работу (последовательность неправильная).

После окончания работы алгоритма стек должен остаться пустым. Если стек не пуст — это означает, что некоторые открывающие скобки не были закрыты, то есть последовательность не является правильной.

**Проверка на правильность последовательности одного вида скобок**

Если последовательность состоит из одного вида скобок, то алгоритм можно упростить так, чтобы он использовать O(1) дополнительной памяти, поскольку в стеке будут храниться только открывающие скобки одного вида. В этом случае достаточно только запоминать размер стека, то есть число открывшихся скобок.

Назовем эту величину ***балансом***. Будем рассматривать скобки по одной с начала скобочной последовательности. Открывающая скобка увеличивает значение баланса на 1, закрывающая — уменьшает. Последовательность будет правильной, если:

1. Величина баланса всегда неотрицательна (во все промежуточные моменты).
2. Баланс всей скобочной последовательности равен 0.

#### Правильные скобочные последовательности. Подсчет количества

## Рекуррентное соотношение для числа правильных скобочных последовательностей

Обозначим число правильных скобочных последовательностей из n пар скобок через Cn.

В частности, C0=1, C1=1, C2=2, C3=5.

Выведем рекуррентное соотношение для числа правильных скобочных последовательностей.

Рассмотрим последовательность из n пар скобок. Найдем в ней скобку, парную первой открывающей скобке. Тогда наша последовательности имеет следующий вид:

**(A)B**

где A и B — также правильные скобочные последовательности. Если скобочная последовательность A состоит из k пар скобок, то такую последовательность можно составить Ck способами, где k=0,…,n−1 (в частности, значение k может быть равно 0, что соответствует пустой  скобочной последовательности). Последовательность B тогда состоит из n−1−k пар скобок, и ее можно составить Cn−1−k способами. Любой скобочной последовательности A можно сопоставить любую скобочную последовательность B, поэтому общее число способов выбрать одну скобочную последовательность A и одну скобочную последовательность B равно CkCn−1−k.

Но также значение k может быть любым от 0 до n−1, поэтому нужно просуммировать результат по k от 0 до n−1:

Cn=C0Cn−1+C1Cn−2+C2Cn−3+…+Cn−1C0=∑k=0n−1CkCn−1−k

## Нахождение числа правильных скобочных последовательностей динамическим программированием

Используя ранее выведенное рекуррентное соотношение, можно найти число правильных скобочных последовательностей методом динамического программирования. Будем в массиве C хранить значения числа правильных скобочных последовательностей:

**C[0] = 1;**

**for (int n = 1; n <= N; ++n)**

**{**

**C[n] = 0;**

**for (int k = 0; k < n; ++k)**

**C[n] += C[k] \* C[n - 1 - k];**

**}**

## Замкнутая формула для числа правильных скобочных последовательностей

Значения Cn в математике называются числами Каталана и возникают они и в ряде других интересных задач.

Для чисел Каталана хорошо известна формула в замкнутном виде:

Cn=C2nnn+1=(2n)!n!(n+1)!

где ССmk=m!k!(m−k)! — число сочетаний из m элементом по k, широко распространенное в комбинаторике выражение.

Легко понять, что количество правильных скобочных последовательностей из двух видов скобок (например, круглых и квадратных) равно Cn2n, из трех видов скобок — Cn3n и т.д.