1、将进出栈表示为+1、-1：如果有n个元素进行进出栈操作，则必须操作2n次，且其中必须含有n次进栈和n次出栈，所以如果通过-1、+1来表示进栈和出栈的操作，那么所有操作的和必然为0；

2、非法序列：通过组合数C（n，2n）可以计算出所有操作的为0的进出栈序列，但是其中混有一些非法的序列，比如说在n=3时，序列：-1，+1，-1，+1，+1，-1，这个序列的第三项是在空栈的时候进行出栈操作，显然是错误的，所以需要排除这类的序列。

3、前缀和：在排除非法序列之前，先引进了一种前缀和的概念，前缀和就是一个长度k序列，前1项、前2项......前k项的和。分析一下，对于上一条中满足所有操作和为0序列，根据栈的性质，一个序列前缀和只要出现小于0的数，则可以判定这个序列是非法序列，所以只需要把这种具有前缀和小于0的序列剔除掉，那么剩下的序列就是合法的进出栈方式。

4、第一个值为-1的前缀和：一个进出栈序列如果出现值小于0的前缀和，就肯定能找到找到第一个值为-1的前缀和，如序列：-1，+1，-1，+1，+1，-1，其中-1、+1、-1这个前缀和就是第一个值为-1的前缀和，通过这个前缀和就可以把序列分为两部分，其中第一部分-1，+1，-1，第二部分+1，+1，-1，第一部分的-1数量比+1多1，第二部分的+1数量比-1多1，此时将第一部分全部取反，得到序列+1，-1，+1，+1，+1，-1，此时在整个序列中+1的个数比-1的个数多了2个。所有的非法序列都符合这个规律，故将其推广。设A是包含长度为2n的非法序列的集合，对A中的所有元素第一个值为-1的前缀和取反。可以得到集合B，B中元素的序列中+1的数量比-1的数量多两个，+1的数量是n+1，-1的数量是n-1，且总能找到一个值为+1的前缀和。

5、对于B中元素的数量是容易计算的，B中的元素要满足两个条件，一是有n+1个+1，n-1个-1，二是总能找到一个值为+1的前缀和，分析一下，实际上只要满足第一个条件就必定满足第二个条件，因为+1的数量总比-1多1（最极端的情况是所有-1都排前面，但这个时候也满足第一个条件），所以只要计算满足第一个条件序列的就可以计算出B中的元素数量，通过组合数C（n+1，2n）就可以算出。

6、现在只要证明A中元素的数量和B中元素的数量相等就可以了，实际上在A中任意元素的第一个值为-1的前缀和取反，都可以在B中找到唯一的一个元素与之对应，同理，在B中对任意的元素的第一个值为+1的前缀和取反，都可以在A中找到唯一的元素与之对应，故A与B中的元素是一一对应的关系，所以A中元素的数量就是C（n+1，2n）。

7、所以最后就是用总的元素数C（n，2n）减去非法序列数C（n+1，2n），最后就可以得到所有的可能进出栈序列数1/（n+1）C（n，2n）。