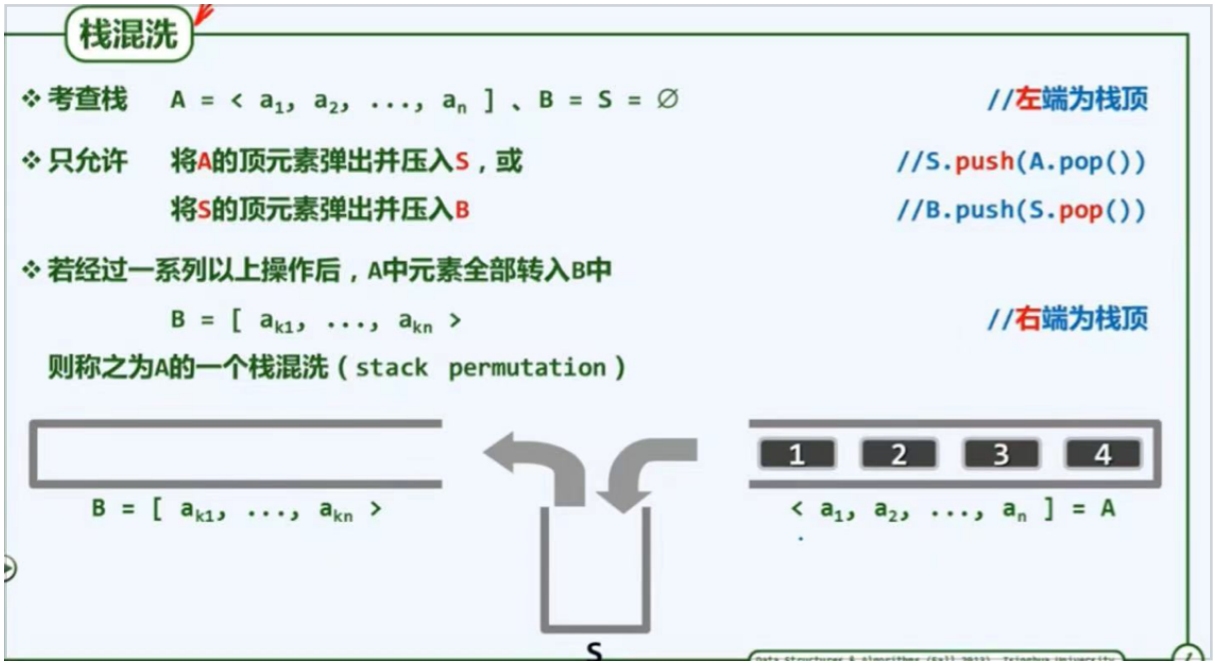
04-E 混洗

#数据结构邓神

什么是栈混洗



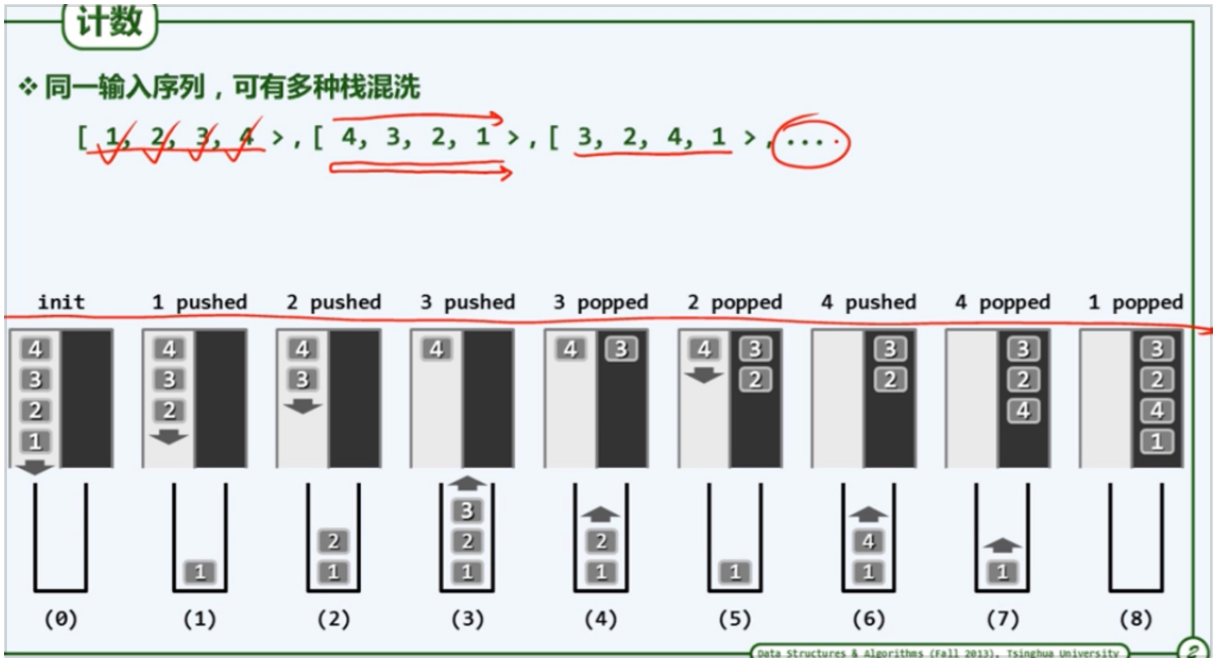
< 表示栈顶部

] 表示栈底部

S 为中转栈

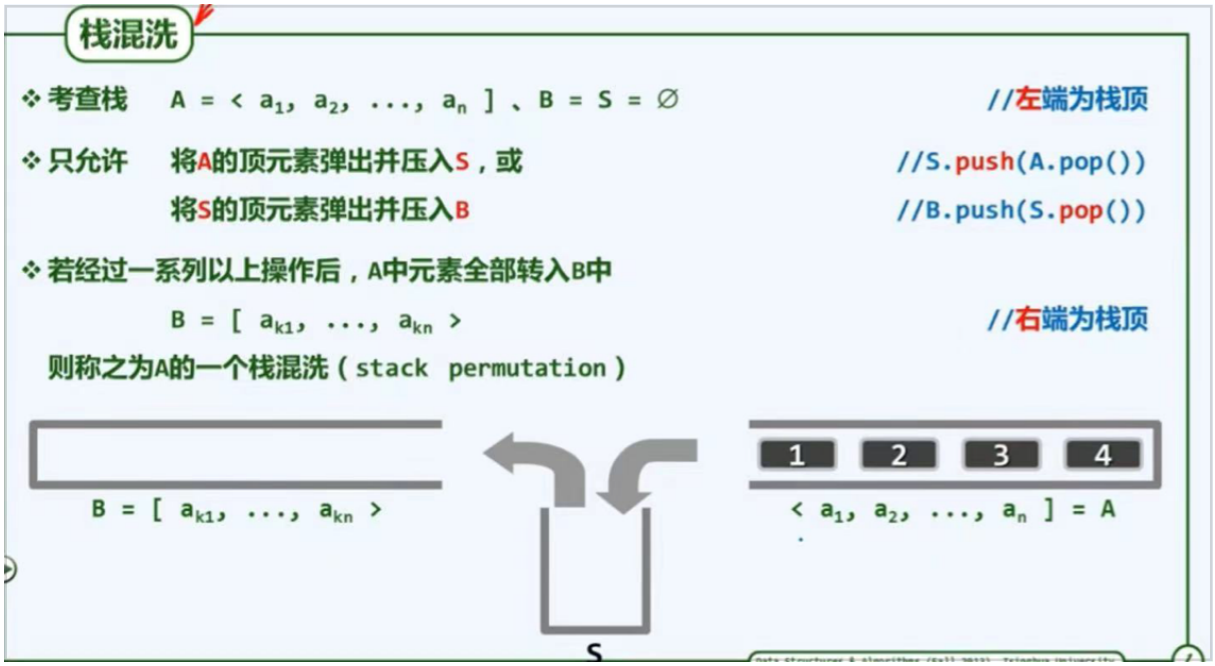
只允许两种操作

1. 弹出A 轧入S
2. 弹出 S 轧入B



对于长度为N的序列，可能的栈混洗总数为多少呢?

SP(N) < N! 一定小于全排列



= catalan(n) = (2n)!/((n+1)!(n)!)

甄别

```
< 1,2,3,...n]
任一排列 [ p1,p2,p3,...,pn>
是否为栈混洗

简单情况 < 1,2,3]
栈混洗 5 种
```

全排列 6 种

缺少的是 [3,1,2>

为什么是他呢?

禁形

充要性

A permutation is a stack permutation iff(if and only if 充分必要条件)

it does NOT involve the permutation

i < j < k

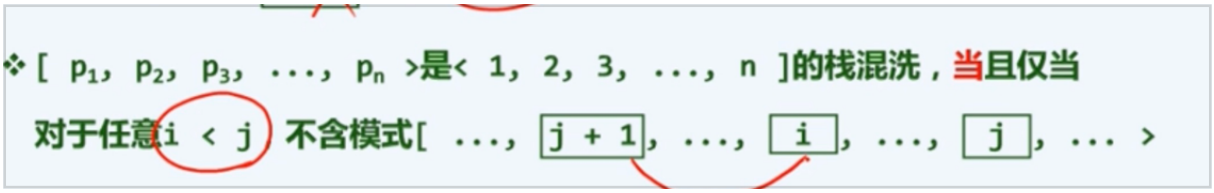
也就是说不含 k i j 这种排列

算法

我们可以直接导出一个 O(n^3)的算法

枚举所有可能

禁形 简化条件



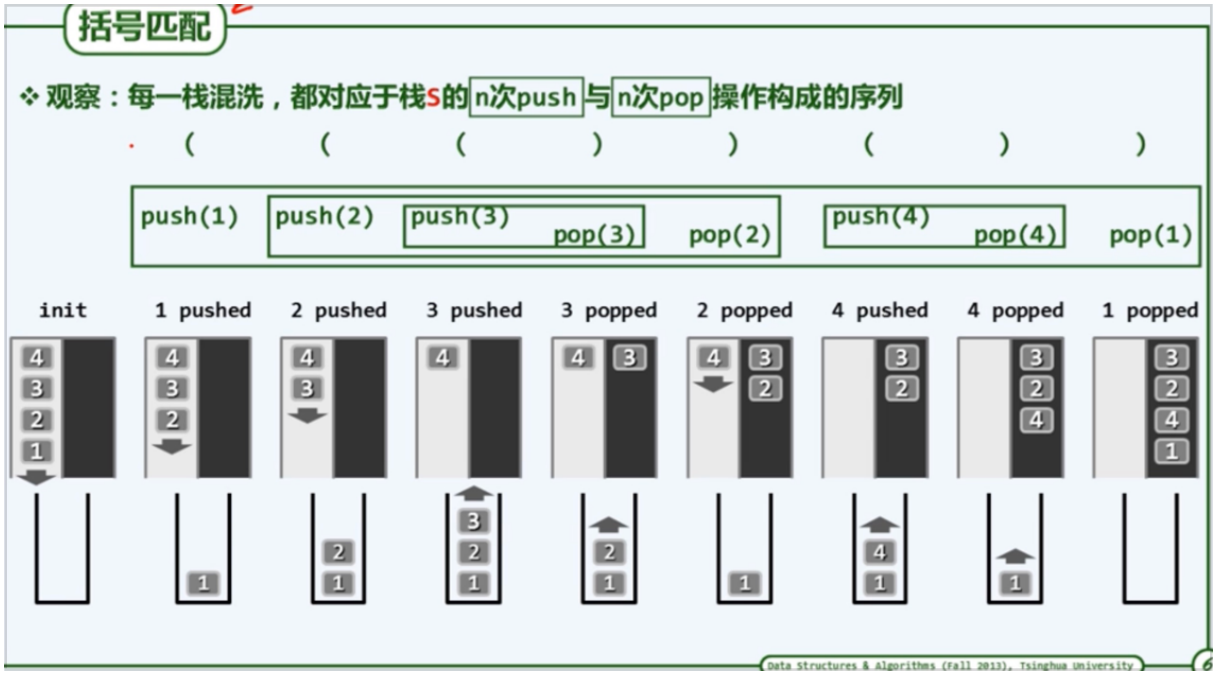
如此可以获得一个O(n^2)的算法

借助栈结构简化为 O(n)


```
bool stackPermutation(stack<int> &A, stack<int> &B) { // 栈混洗 O(n) 模拟算法
    stack<int> S, temp;
    while (!B.empty()) {
        temp.push(B.top());
```

```
            B.pop();
        }
        while (!A.empty()) {
            S.push(A.top());
            A.pop();
            if (temp.top() == S.top()) {
                temp.pop();
                S.pop();
                while (!S.empty()) {
                    if (temp.top() == S.top()) {
                        temp.pop();
                        S.pop();
                    }
                    else return false;
                }
            }
        }
        return S.empty();
    }
```

括号匹配 与 栈混洗



合法的栈混洗系列和合法的括号匹配表达式右一一对应的关系