

2022.9.18.
数学作业三.

1.
基本思想就是利用另一个栈将一个栈内元素倒置排列,
使得可以先出栈队头元素, S_2 栈顶视为队头

```
struct queue {  
public:  
    enqueue(Node x);  
    dequeue(); // 返回 false 为删除失败  
    queue_empty();  
private:  
    stack s1, s2;  
};
```

对于函数 enqueue:
Input: Node x
Output: void

push(S_1, x) 即可
函数 dequeue:

```
Output: bool  
if empty( $S_2$ ) then:  
    while (!empty( $S_1$ )) // 将  $S_1$  中元素倒序放入  $S_2$ .  
        Node tmp;  
        pop( $S_1, tmp$ );  
        push( $S_2, tmp$ );  
    Node front;  
    pop( $S_2, front$ );  
    return true;  
else:  
    Node front;  
    pop( $S_2, front$ );  
    return true;  
函数 queue_empty:
```

```
return empty( $S_1$ ) && empty( $S_2$ );
```

2.
设 $f(x)$ 为 x 辆火车出站顺序的可能数

设最后一个出站的火车为第 k 辆

则 $f(x, k) = f(k-1) \cdot f(x-k)$

从而 $f(x) = \sum_{k=1}^x f(k-1) f(x-k)$

此递推式为 Catalan 数的递推式

由于 $f(0) = f(1) = 1$, 因此 $f(x)$ 即为 Catalan 数 C_x

从而 $f(x) = \frac{C_{2n}}{n!}$

3.

必要性:

假设存在 i, j, k s.t. 满足 $i < k < j$ 且 $P_j < P_k < P_i$.

根据出栈顺序 P_i 出栈时 P_j, P_k 均未出栈

又考虑到 P_i 为最大的, 则此时 P_j 与 P_k 必定在栈内

且由于 $P_j < P_k$, 则 P_j 更靠近栈底

在出栈到 P_j, P_k 时, 显然先出栈 P_k , 从而出栈顺

序必为 $i < k < j$, 与 $i < j < k$ 矛盾.

因此假设不成立, 必要性得证.

充分性

考虑构造一个入栈方式使得出栈顺序为 P_1, P_2, \dots, P_n

① 入栈 P_1 次, 再出栈 1 次, 得到 P_1 ,

② 若 $P_2 > P_1$, 则入栈 $P_2 - P_1$ 次, 出栈 1 次, 得到 P_2

若 $P_2 = P_1 - 1$, 直接出栈, 得到 P_2 .

若 $P_2 < P_1 - 1$, 则存在 $1 < 2 < m$ s.t. $P_2 < P_m < P_1$

与题设矛盾, 从而不存在此情况 (若存在便不可构造出入栈方式 s.t. 出栈顺序为 P_1, P_2, \dots, P_n)

造出入栈方式 s.t. 出栈顺序为 P_1, P_2, \dots, P_n)

③ 重复②中操作直至出栈完成