

2023数算第三章作业

1.

请利用两个栈S1和S2来模拟一个队列。已知栈的三个运算定义如下：

push(ST, x): 元素x入ST栈；

pop(ST, x): ST栈顶元素出栈，赋给变量x；

empty(ST): 判ST栈是否为空。

那么如何利用栈的运算来实现队列的三个运算：enqueue: 插入一个元素入队列； dequeue: 删除一个元素出队列； queue_empty: 判队列是否为空。（请写明算法的思想及必要的注释）。

因为有两个栈，所以可以这么做：

enqueue时，把新元素全部放到 S1 栈顶。

dequeue时，要访问栈 S1 的栈底，可以把 S1 栈顶的元素pop掉，push进 S2 的栈顶，反复执行直到 S1 变成空栈。此时 S2 栈顶的元素即为原来栈底的元素。处理完后把 S2 的元素放回去就行了。

queue_empty时，返回empty(S1)即可。

如果用代码来写的话，是这样的：

```
void enqueue(int x) {
    push(S1, x);
}

void dequeue(int &x) {
    int ret;
    while (!empty(S1)) {
        int t;
        pop(S1, t);
        push(S2, t);
    }
    pop(S2, ret);
    while (!empty(S2)) {
        int t;
        pop(S2, t);
        push(S1, t);
    }
    return ret;
}

bool queue_empty() {
    return empty(S1);
}
```

2.

编号为 $1, 2, \dots, n$ 的 n 辆火车顺序开进栈式结构的站台。请问开出车站的顺序有多少种可能？请写出你的推导过程。

如果把入栈操作记为 $($ ，把出栈操作记为 $)$ ，则可知每一种可能的出入栈顺序与合法的括号串一一对应。

另外，容易证明，若出入栈顺序不同，则它们对应的火车开出车站的顺序也不同。证法是考虑第一个位置不同的左括号。

设答案为 C_n ，则它满足递推式：

$$\begin{aligned}C_0 &= 0 \\C_1 &= 1 \\C_n &= \sum_{i+j=n-1} C_i C_j\end{aligned}$$

显然， $\{C_n\}$ 为Catalan数，答案为

$$\binom{2n}{n} - \binom{2n}{n-1}$$

3.

证明：从初始输入序列 $1, 2, \dots, n$ ，可以利用一个栈得到输出序列 P_1, P_2, \dots, P_n （ P_1, P_2, \dots, P_n 是 $1, 2, \dots, n$ 的一种排列）的充分必要条件是：不存在下标 i, j, k ，满足 $i < j < k$ 同时 $P_j < P_k < P_i$ 。

必要性：显然。首先，容易证明栈内始终有序。因为在 pop 出 P_i 后，再 pop 出 P_j 肯定会把介于 P_i 和 P_j 的所有数都 pop 出来，因为 $P_j < P_k < P_i$ ，则它也在 P_j 之前被 pop 了出来，即 $i < k < j$ 。

充分性：构造法。我们通过模拟来构造一种合法的出入栈序列。具体来说，是维护一个栈 S ，并用 i 遍历 1 到 n ，进行如下操作

1. 如果 P_i 尚未入栈，那么把输入序列中尚未入栈的元素逐一入栈，到 P_i 入栈为止。
2. 断言现在栈顶的元素是 P_i ，把 P_i 出栈。

我断言在这个过程中一定没有问题。

不然，假设在模拟的过程中出了问题，则必定是在第二步出现的。设出问题时，尝试取出的元素为 P_j 。

此时 P_j 一定在栈里面（否则会更早出问题），且由栈的单调性，栈顶是一个比 P_j 大的数，不妨设为 P_k 。

设 P_k 一定是在对某个数执行构造算法中的 1. 的时候放进去的，设它为 P_i ，则我们有： $P_j < P_k < P_i$ 。

考虑 i, j, k 的大小关系。因为在执行 P_i 出栈时， P_j 尚未出栈，所以有 $i < j$ 。

又因为在尝试执行 P_j 出栈时， P_k 尚不需要出栈，所以有 $j < k$ 。

因此，我们构造出来了一组下标 i, j, k ，满足 $i < j < k$ 同时 $P_j < P_k < P_i$ ，与题设矛盾。

因此，这个算法一定不会出问题。