1

存储数据大小为 n

```
int top(){
    int res;
    while(!A.empty()){
        int tmp =  A.front();
        A.pop_front();
        B.push_back(tmp);
        if(A.empty())
            res = tmp;
    }
    while(!B.empty()){
        A.push_back(B.front());
        B.pop_front();
    }
    return res;
}
```

时间复杂度 O(n)

```
void pop(){
    while(!A.empty()){
    int tmp = A.front();
    A.pop_front();
    if(!A.empty())
        B.push_back(tmp);
    }
    while(!B.empty()){
        A.push_back(B.front());
        B.pop_front();
    }
}
```

时间复杂度 O(n)

```
void push(int &x){
    A.push_back(x);
}
```

时间复杂度 O(1)

```
bool empty(){
    return A.empty();
}
```

时间复杂度 O(1)

2

formula

出栈顺序等价于求 n 个 0，n 个 1 组成的排列个数。
该排列满足性质，对于任意其任意子排列 [0,a), 该子排列满足 0 的个数大于等于 1
总的排列数 C_{2n}^{n}
求不满足要求的排列数
对于任何不满足的排列，存在一个子排列，使得子排列有 $x+1$ 个 1，x 个 0，然后将改子排列的 0，1 互换，就变成了 n+1 个 0，n-1 个 1 的排列，且这种转换一一对应。个数为 C_{2n}^{n+1}
Catalan 数 =$C_{2n}^{n} - C_{2n}^{n+1} = \frac{1}{n}C_{2n}^{n}$

Prove

这道题等价于求当 $i < j < k$, 所有可能出现的 P_i, P_j, P_k 的顺序
1. j 入栈时，i 已经出栈
1.1 j 出栈后，k 入栈, 出站顺序 i，j,k $P_i < P_j < P_k$
1.2 k 先入栈，出站顺序 i，k，j$P_i < P_k < P_j$
2 j 入栈时，i 未出栈，同时先于 k 入栈出栈
2.1 k 先于 i 出栈出栈顺序 j,k,i $P_k < P_i < P_j$
2.2 k 后于 i 出栈出栈顺序 j,i,k $P_j < P_i < P_k$
3 k 入栈，i,j 都没出栈，出栈顺序 k,j,i $P_k < P_j < P_i$
综上，对于 P_i, P_j, P_k 的排列，只有 $P_j < P_k < P_i$ 没有出现
所以不存在下表 i, j, k，使得 $P_j < P_k < P_i$