Coding-2022-4-3

-PROMISE-

Sunday 3rd April, 2022

1 两个栈实现一个队列

总的来说很简单，这里没有用 stack 而是自己写了数组实现的;

```
class CQueue {
        private:
        int* s1;
        int* s2;
        int head1, tail1;
        int head2, tail2;


        public:
        CQueue() {
                s1=new int[10005];
                s2=new int[10005];
                head1=0;
                head2=0;
                tail1=0;
                tail2=0;
        }
        bool add(int val){
                if(tail1>=10005){
                        return false;
                }
```

```
22                  s1[ tail1 ]=val;
23                  tail1 ++;
24                  return  true;
25          }
26
27          void  move(){
28                  head2=0;
29                  tail2 =0;
30                  for( int  i=tail1-1;i>=0;i--){
31                          s2[ tail2 ++]=s1[i];
32                  }
33          }
34
35          int  cdelete (){
36                  if ( tail2 <1){
37                          return  -1;
38                  }
39                  tail2 --;
40                  head1=0;tail1=0;
41                  for( int  j=tail2-1;j>=0;j--){
42                          s1[ tail1 ++]=s2[j];
43                  }
44                  return  s2[ tail2 ];
45          }
46
47          void  appendTail( int  value ) {
48                  if (add(value)==false){
49                          cout<<"false"<<endl;
50                  }
51
52          }
53
54          int  deleteHead() {
```

```
55                  move();
56                  return  cdelete ();
57          }
58  };
```

2 offer 30: 包含 min 的栈

要求是除了常规的 top，pop，push 等函数，需要实现输出栈中最小值的 min 函数

选择是维护一个最小栈，但实际上有所浪费了，辅助栈中并不需要记录所有的值，而是只有栈为空或者插入元素小于自己的栈顶元素才插入，这里贴官方解答，最好是自己实现栈

```
1   class  MinStack {
2           public:
3           stack<int> st;
4           stack<int> minStack;
5
6           MinStack() {
7                   while(!st.empty()) {
8                           st.pop();
9                   }
10                  while(!minStack.empty()) {
11                          minStack.pop();
12                  }
13                  minStack.push(INT_MAX);
14          }
15
16          void  push(int  x) {
17                  st.push(x);
18                  int  minVal = std::min(minStack.top(), x);
19                  minStack.push(minVal);
20          }
21
```

```
22          void pop() {
23                  st.pop();
24                  minStack.pop();
25          }
26
27          int top() {
28                  return st.top();
29          }
30
31          int min() {
32                  return minStack.top();
33          }
34  };
```