

# Homework 6

---

罗海旻 U201514716

1. Can you implement a version of stack that does not use the bottom field in the struct stack\_header ?

答：仅使用栈顶(指向下一个未使用的位置)，即当前元素个数实现即可。

2. Consider what would happen if we pop an element from the empty stack when contracts are not checked? When does an error arise?

答：可能会导致数组访问越界。在对正好为空的栈pop两次时发生错误，第一次仅返回bottom处的元素，第二次栈顶指针变为负数，导致越界。

3. Our queue implementation wastes a lot of space unnecessarily. After enqueueing and dequeuing a number of elements, the back may reach the capacity limit. If the front has moved on, then there is a lot of space wasted in the beginning of the data array. How can you change the implementation to reuse this storage for enqueueing elements? How do you need to change the implementation of enq and deq for that purpose?

答：实现循环队列。

4. Our queue design always “wasted” one element that we marked X. Can we save this memory and implement the queue without extra elements? What are the tradeoffs and alternatives when implementing a queue?

答：使用当前队列长度即可，这使得需要将队尾标志更改为长度标志，且需要经过计算才能确定队尾位置。

5. The stack implementation using arrays may run out of space if its capacity is exceeded. Can you think of a way of implementing unbounded stacks stored in an array?

答：栈超过一定大小时进行扩容，即开辟一个大小更大的数组(原数组两倍)，随后将原数组内容拷贝到新的数组。