Tutorial (week 5)

All the answers can be given with pseudocode or with Python. Remember that most of the time, many solutions are possible. Note that certain classes defined in the solutions are not predefined in Python. For example, `Stack()` is not a predefined Python class.

1. Suppose you have access to an empty stack `S`. Write the content of `S` after execution of each of these commands:

```
S.push(4)
S.push(6)
S.pop()
S.push(8)
S.push(2)
S.push(1)
S.pop()
S.pop()
S.pop()
```

2. Same question than before, but this time with a queue `Q`:

```
Q.enqueue(4)
Q.enqueue(6)
Q.dequeue()
Q.enqueue(8)
Q.enqueue(2)
Q.enqueue(1)
Q.dequeue()
Q.dequeue()
Q.dequeue()
```

3. Suppose you are given an array, A, containing n numbers in order. Describe in Python or pseudocode an efficient algorithm for reversing the order of the numbers in A using a single for-loop that indexes through the cells of A, to insert each element into a stack, and then another for-loop that removes the elements from the stack and puts them back into A in reverse order. What is the running time of this algorithm?

4. Solve the previous exercise using a queue instead of stack. That is, suppose you are given an array, A, containing n numbers in order, as in the previous exercise. Describe in Python or pseudocode an efficient algorithm for reversing the order of the numbers in A using a single for-loop that indexes through the cells of A, to insert each element into a queue, and then another for-loop that removes the elements from the queue and puts them back into A in reverse order. What is the running time of this algorithm?

5. Describe how to implement a queue using two stacks, so that the amortized running time for dequeue and enqueue is $O(1)$, assuming that the stacks support constant-time push, pop, and size methods. What is the worst-case running time of the `enqueue()` and `dequeue()` methods in this case ?

6. Describe how to implement a stack using two queues. What is the running time of the `push()` and `pop()` methods in this case ?

Hints

- **Question 3:** think about the order of the indexing for each of the for-loops.
- **Question 4:** think about the order of the indexing for each of the for-loops.
- **Question 5:** use one stack for enqueues and the other for dequeues (you still need to say how and you also need to do the amortized analysis).
- **Question 6:** use one queue as auxiliary storage and keep track of sizes as you are using it.