

1. Suppose we have implemented the Queue ADT as a singly-linked-list with head and tail pointers and no sentinels. Which of the following best describe the tightest running times for the functions enqueue and dequeue, assuming there are $\Theta(n)$ items in the list, and that the front of the queue is at the head of the list?

- A. $\Theta(1)$ for enqueue and $\Theta(n)$ for dequeue.
- B. **[Correct Answer]** **[Your Answer]** $\Theta(1)$ for both.
- C. $\Theta(n)$ for enqueue and $\Theta(1)$ for dequeue.
- D. $\Theta(n)$ for both.
- E. None of the options is correct

2. Suppose we have implemented the Stack ADT as an array. Every time the array is full, you resize the array by creating a new array that can hold one element more than the previous array and copy values over from the old array. What is the total running time for n pushes to the stack.

- A. $\Theta(1)$.
- B. $\Theta(n)$.
- C. $\Theta(n \log n)$.
- D. $\Theta(\log n)$.
- E. **[Correct Answer]** **[Your Answer]** $\Theta(n^2)$.

3. Objects of type iterator promise to implement each of the following except:

- A. **[Your Answer]** All of these are implemented in an iterator.
- B. **[Correct Answer]** operator+
- C. operator==
- D. operator*
- E. operator=

4. Suppose `queue<int> q` contains 6 elements 1, 2, 3, 4, 5, 6 (enqueued in that order). What is the result of executing the following code snippet? (Assume member function `front()` returns the value found at the front of the queue without removing it.)

```
for(int i = 1; i<7; i++){
    if(i%2==0) {
        q.enqueue(q.front());
        q.dequeue();
    }
}
```

- A. The even numbers in `q` are reversed.
- B. The elements `q` are reversed.
- C. **[Correct Answer]** **[Your Answer]** The front half of the original `q` is now at the back half.
- D. `q` remains the same.
- E. The odd numbers in `q` are reversed.

5. In implementing Stack ADT, using which of the following data structure gives worst asymptotic runtime for push? (Assume we require to push at the end of list or array)

- A. **[Correct Answer]** Singly linked list with head pointer only.
- B. **[Your Answer]** Array (size of array larger than possible elements in stack).
- C. Doubly linked list with head and tail pointer.
- D. All options provide the same runtime.
- E. Singly linked list with head and tail pointer.