NetID: zeruiw2 QuizID: 98297 Score: 3 / 5 Answer Source: PrairieLearn

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 $O(n)$ items in the list, and that the rear of the queue is at the `head` of the list?

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

2. Consider this (and assume all STL classes are available):

```
struct animal {
    string name;
    string food;
    animal(string n="blob", string f="you"): name(n), food(f) { }
};

int main() {
    animal g("giraffe","leaves"), b("bear");
    list<animal *> zoo;
    zoo.push_back(&g); zoo.push_back(&b); //STL list insertAtEnd
    for(list<animal *>::iterator it = zoo.begin(); it != zoo.end(); it++)
        cout << (*it)->name << " " << (*it)->food << endl;
}
```

Which of the following describes the result when this function is compiled and run?

A. There is a compiler error because of a type mismatch in the parameterized type for the list class.
B. There is a runtime error because the iterator is dereferenced twice.
C. There is a compiler error because there is no constructor matching the one called for variable b.
D. [Correct Answer] [Your Answer] The code sends "giraffe leaves" and "bear you" to standard out.
E. The code sends "giraffe leaves" to standard out.

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

4. We have implemented the Stack ADT as an array. Every time the array is full, you resize the array creating a new array that can hold four times as many elements as the previous array and copy values over from the old array. What is the total running time for n pushes to the stack.

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

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

A. Singly linked list with head pointer only
B. [Correct Answer] [Your Answer] Exactly two of the other options are correct
C. Doubly linked list with head pointer only
D. Singly linked list with head and tail pointer
E. Doubly linked list with head and tail pointer only