CS 201, Fall 2014 — Nov 19th — Exam 2 Name: Partial solution

Note: In questions where you are asked about a static method, assume that the method is in a class called `Qn` where n is the question number, e.g., `Q1` for Question 1.

Question 1. [10 points] Consider the following code:

```
int count = 0;
for (int i = 0; i < n*n; i++) {
    count++;
}
for (int j = 0; j < n; j++) {
    count++;
}
```

N^2 times; O(1); N times; O(1)

low order term dropped

$N^2 \cdot O(1) + N \cdot O(1)$ is $O(N^2)$

State a big-O upper bound on the running time of this code, using n (the value of the variable `n`) as the problem size. Briefly explain your answer.

Question 2. [10 points] Consider the following code:

```
int count = 0;
for (int i = 0; i < n; i++) {
    for (int j = 0; j < i/2; j++) {
        count++;
    }
}
```

n times; average # of iterations is n/4; O(1)

State a big-O upper bound on the running time of this code, using n (the value of the variable `n`) as the problem size. Briefly explain your answer.

$n \cdot \frac{n}{4} \cdot O(1)$ is $O(n^2)$

Question 3. [10 points] Consider the following method:

```
public static<E> void removeEveryOther(List<E> list) {
    Iterator<E> i = list.iterator();
    int count = 0;
    while (i.hasNext()) {
        i.next();
        if (count % 2 == 1) {
            i.remove();
        }
        count++;
    }
}
```

Note that an `Iterator`'s `remove` method removes from the collection the last element returned by the `Iterator`'s `next` method.

(a) State a big-O upper bound on the running time of this method if `list` is an `ArrayList`. Consider the problem size N to be the number of list elements. Explain briefly.

Iterator Remove() takes time proportional to the number of elements that must be shifted, which is

$(N-2), (N-4), (N-6), \ldots, 2, 0$

the sum of this series is $O(N^2)$, so the overall running time is $O(N^2)$

(b) State a big-O upper bound on the running time of this method if `list` is a `LinkedList`. Consider the problem size N to be the number of list elements. Explain briefly.

Iterator remove() is always $O(1)$ for a LinkedList, so $O(N)$ overall