CS 201, Summer 2015 — July 7th — Exam 2 Review Quiz Name: Solution

Question 1. [25 points] State a big-O upper bound on the running time of the following method, where the problem size N is the number of elements in the array passed as the method's parameter. Explain your answer briefly.

N = arr.length

```
public static int q1(int[] arr) {
  int sum = 0;
  for (int i = 0; i < arr.length; i++) {      — N times
    for (int j = 0; j < arr.length; j++) {    — N times
      sum += (arr[i] * arr[j]);               O(1)
    }
  }
  return sum;
}
```

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

Question 2. [25 points] State a big-O upper bound on the running time of the following method, where the problem size N is the number of elements in the array passed as the method's parameter. Explain your answer briefly.

N = arr.length

```
public static int q2(int[] arr) {
  int sum = 0;
  for (int i = 0; i < arr.length; i++) {
    int max = arr.length;
    if (max > 10000) {
      max = 10000;
    }
    for (int j = 0; j < max; j++) {           — O(1)
      sum += (arr[i] * arr[j]);               — O(1)
    }
  }
  return sum;
}
```

max is at most 10,000, thus $O(1)$

$N \cdot O(1) \cdot O(1)$ is $O(N)$

Question 3. [25 points] State a big-O upper bound on the running time of the following method, where the problem size N is the number of elements in the `ArrayList` passed as the method's parameter. Explain your answer briefly.

```
public static int q3(ArrayList<Integer> arr) {
  int sum = 0;
  for (int i = 0; i < arr.size(); i++) {
    sum += arr.get(i);
  }
  return sum;
}
```

N = arr.size()

N times

O(1)

O(1)

$N \cdot O(1)$ is $O(N)$

Question 4. [25 points] State a big-O upper bound on the running time of the following method, where the problem size N is the number of elements in the `LinkedList` passed as the method's parameter. Explain your answer briefly.

```
public static int q4(LinkedList<Integer> arr) {
  int sum = 0;
  for (int i = 0; i < arr.size(); i++) {
    sum += arr.get(i);
  }
  return sum;
}
```

N = arr.size()

O(1)

N times

time is proportional to number of elements that must be skipped to reach the element at index i — average is N/2

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