

# CPSC 131 Homework 2 solutions

Max points: 9

## #1

Choose the equivalent Big O notation for the given functions. **[1 point each]**

1.  $2n + 3$   
 $O(1)$   
 **$O(n)$**   
 $O(n^2)$   
 $O(\log n)$
2.  $n^2 + n + 1$   
 $O(1)$   
 $O(n)$   
 **$O(n^2)$**   
 $O(\log n)$
3.  $n^2 + \log n$   
 $O(1)$   
 $O(n)$   
 **$O(n^2)$**   
 $O(\log n)$

## #2

Choose the Big O complexity for each of the given pieces of code. Give a 1-2 sentence explanation of your reasoning. **[2 points each]**

1.

```
sum = 0;
for (int i=0; i<n; i++) {
    sum = sum + numbers[i];
}
```

- a.  $O(1)$
- b.  $O(n)$**
- c.  $O(n^2)$
- d.  $O(\log n)$

*The code has to go through the loop exactly  $n$  times.*

2.

```
bool containsDuplicates(int numbers[], int n) {
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            if (i == j) // Don't compare with self
                continue;
            if (numbers[i] == numbers[j])
                return true;
        }
    }
    return false;
}
```

- a.  $O(1)$
- b.  $O(n)$
- c.  $O(n^2)$
- d.  $O(\log n)$

*There are two nested loops, each of length  $n$ . Therefore, the if statements in the inner loop will be executed  $n*n$  times.*

3.

```
sum = 0; sum2 = 0;
for (int i=0; i<n; i++) {
    sum = sum + numbers[i];
}
for (int j=0; j<n; j++) {
    sum2 = sum2 + numbers[j];
}
```

- a.  $O(1)$
- b.  $O(n)$
- c.  $O(n^2)$

*The two loops are not nested but are executed one after the other. Therefore, the summations will happen  $2*n$  which is also  $O(n)$ .*