Problem Set 1 **CS 3851** **Name** Hudson Amey

Submission instructions: Your written answers should be submitted to Canvas as a PDF.

Assumption array is not empty

1. Write the equation T(n) modelling the run time in terms of the size n of the input array.

```
public static int findMinimum(int[] input) {
1    int minimum = Integer.MAX_VALUE;
2    for(int i = 0; i < input.length; i++) {
3        if(input[i] < minimum) {
4            minimum = input[i];
5        }
6    }
7    return minimum;
}
```

c	#t
c_1	1
c_2	1
c_3	$n+1$
c_4	n
c_5	n
c_6	x
c_7	1

$1+1+n(1+1+1+1)+1+1$ where $x=n$ worst

$= 4n+4$

$= 4(n+1)$

$T(n) = c_1 + c_2 + (n+1)c_3 + nc_4 + nc_5 + xc_6 + c_7$

$T(n) = O(n)$

Best $= (c_3+c_4+c_5)n + c_1+c_2+c_3+c_7$

$3n+4$

$x=1$

[max, ..., min]

2. Write the equation T(n) modelling the run time in terms of n.

```
public static long fibonacci(int n) {
1    int a = 0;
2    int b = 1;
3    for(int i = 0; i < n; i++) {
4        int c = a + b;
5        a = b;
6        b = c;
7    }
8    return a;
}
```

$1+1+1+n(1+1+1+1+1)+1+1$

$5n+5$

$5(n+1)$

$T(n) = O(n)$

3. Write the equation T(n, m) modelling the run time in terms of n and m.

```
1 for(int i = 0; i < n; i++) {
2     for(int j = 0; j < m; j++) {
3         int result = i * j;
4         System.out.println(i + " x " + j + " = " + result);
5     }
6 }
```

$1+n(1+m(1+1+1+1)+1)+1$

$2+n(4m+3)$

$4mn+3n+2$

$T(n)=O(n*m)$

4. Write the equation T(n) modelling the run time in terms of the size n of the input array.

```
 public boolean contains(int[] array, int key) {
1     boolean found = false;
2     for(int i = 0; i < array.length; i++) {
3         if(array[i] == key) {
4             found = true;
5         }
6     }
7     return found;
 }
```

$1+1+n(1+1+1+1)+1+1$

$4n+4$

$4(n+1)$

$O(n)=T(n)$

5. Write the equation T(n) modelling the run time in terms of the size n of the linked list.

```
public boolean contains(Node<Integer> head, Integer key) {
1     boolean found = false;
2     Node<Integer> current = head;
3     while(current != null) {
4           if(current.datum.equals(key)) {
5                 found = true;
6           }
7           current = current.next;
8     }
9     return found;
}
```

$1+1+n(1+1+1+1)+1+1$

$4n+4$

$4(n+1)$

$O(n) = T(n)$

6. Write the equation T(n) modelling the run time in terms of the index n.

```
public String get(String[] array, int index) {
1     if(index >= array.length || index < 0) {
2           return null;
3     }
4     return array[index];
}
```

Worst: $1+1+1 = 3 = O(3)$

$= O(1) = T(1)$

7. Write the equation T(n) modelling the run time in terms of the size n of the linked list.

```
 public String get(Node<String> head, int index) {
1      int currentIndex = 0;
2      Node<Integer> current = head;
3      while(current != null && currentIndex <= index) {
4            if(currentIndex == index) {
5                  return current.datum;
6            }
7            current = current.next;
8            currentIndex++;
9      }
8      return null;
 }
```

$1+1+n(1+1+1+1+1)+1+1$

$6n+4$

$2(3n+2)$

$O(n) = T(n)$ → If LL is not empty

Multiple returns, $O(1)$ if LL is empty