

## 1 Written Exercises

1. Consider the following classes of functions:

- $O(n)$
- $O(\log(n))$
- $O(n^2)$
- $O(1)$
- $O(2^n)$
- $O(n^3)$
- $O(n \cdot \log(n))$

Say each of the above functions maps from input size  $n$  to the *approximated* algorithm running time. Sort, from left to right, the above classes of functions from the cheapest to the most expensive.

**Caution:** You will lose **all** marks if the order is not completely correct.

[ of 10 marks]

2. Consider the following statements:

- (A)  $3n + 7$  is  $O(n \cdot \log(n))$
- (B)  $3n + 7$  is  $O(n)$
- (C)  $3n + 7$  is  $O(1)$
- (D)  $3n + 7$  is  $O(2^n)$
- (E)  $3n + 7$  is  $O(\log(n))$
- (F)  $3n + 7$  is  $O(n^2)$

(a) Which of the above statement or statements are *correct*?

[ of 10 marks]

(b) Among the above statement or statements that are *correct*, which one is the most *accurate*?

[ of 5 marks]

(c) Justify your answer to the previous question. That is, clearly explain why it is more *accurate* than all other *correct* statements.

[ of 10 marks]

3. In order to prove that  $f(n) = 4n^3 - 5n^2 + 59 + n^4 + 9n$  is  $O(n^4)$ , you need to choose values for two constants: constant  $c$  as a factor for  $n^4$  and constant  $n_0$  as some starting value of  $n$ .

(a) Write down the precise condition for which  $c$  and  $n_0$  must satisfy in order for the proof to succeed. **Hint:** Your answer should involve  $n^4$ ,  $f(n)$ ,  $c$ , and  $n_0$ .

[ of 5 marks]

(b) Give values of  $c$  and  $n_0$  that will complete the proof.

[ of 5 marks]

4. Consider the following Java program:

```
1 void prog(int[] a, int n)
2   for (int i = 0; i < n; i++) {
3     for (int j = i; j < n; j++) {
4       for (int k = j; k > 0; k--) {
5         System.out.println(i * j + k);
6       }
7     }
8   }
```

Determine the **most accurate** asymptotic upper bound of the above program, using the big-Oh notation. You **must** show in detail how you determine the bound. Without a convincing derivation process, you will only receive partial marks.

[      of 15 marks]

5. Consider the following Java code:

```
1 boolean isSorted(int[] a) {
2     return isSortedHelper(a, 0, a.length - 1);
3 }
4 boolean isSortedHelper(int[] a, int from, int to) {
5     if (from > to) {
6         return true;
7     }
8     else if (from == to) {
9         return true;
10    }
11    else {
12        return a[from] <= a[from + 1]
13            && isSortedHelper(a, from + 1, to);
14    }
15 }
```

Prove, via mathematical induction, that the method `isSorted` method above correctly returns `true` if the array `a` is sorted in a non-descending order; and `false` otherwise.

[      of 20 marks]