CS 457, Data Structures and Algorithms I
First Problem Set

September 26, 2018

Due on October 7. Collaboration is allowed; please mention your collaborators.

1. (16 pts) Find two functions $f(n)$ and $g(n)$ that satisfy the following relationship. If no such f and g exist, then try to explain why this is the case.

   a) $f(n) = o(g(n))$ and $f(n) \notin \Theta(g(n))$
   b) $f(n) = \Theta(g(n))$ and $f(n) = o(g(n))$
   c) $f(n) = \Theta(g(n))$ and $f(n) \notin O(g(n))$
   d) $f(n) = \Omega(g(n))$ and $f(n) \notin O(g(n))$

2. (16 pts) For each of the following questions, briefly explain your answer.

   a) If I prove that an algorithm takes $O(n^2)$ worst-case time, is it possible that it takes $O(n)$ time on some inputs?
   b) If I prove that an algorithm takes $O(n^2)$ worst-case time, is it possible that it takes $O(n)$ time on all inputs?
   c) If I prove that an algorithm takes $\Theta(n^2)$ worst-case time, is it possible that it takes $O(n)$ time on some inputs?
   d) If I prove that an algorithm takes $\Theta(n^2)$ worst-case time, is it possible that it takes $O(n)$ time on all inputs?

3. (10 pts) Let $f(n)$ and $g(n)$ be any two functions with $f(n) > 1$ and $g(n) > 1$ for all n. Is it true that $15f(n)^2 + 23g(n)^2 = \Theta((f(n) + g(n))^2)$? Give a formal justification for your answer.

4. (15 pts) Is the following statement **true** or **false**, and why? (provide a formal argument either way)

   If $f(n) = O(g(n))$ then $2^{f(n)} = O(2^{g(n)})$.

5. (20 pts) What value is returned by the following function? Express your answer as a function of n. From this, deduce the worst-case running time using Big Oh notation. For full credit, provide a lower bound as well, thus leading to a bound with $\Theta(\cdot)$ notation. Then, also deduce the worst-case running time if we change the while loop clause to $j \leq i$ instead of $j \leq n$.

```
1 function foo1(n)
2 r := 0
3 for i := 1 to n do
4     j := 1
5     while j ≤ n do
6         r := r + 1
7         j := 2 · j
8 return r
```

6. (14 pts) What value is returned by the following function? Express your answer as a function of n. From this, deduce the worst-case running time using Big Oh notation. For full credit, provide a lower bound as well, thus leading to a bound with $\Theta(\cdot)$ notation.

```
1 function foo2(n)
2 r := 0
3 j := n
4 while j ≥ 1 do
5     for i := 1 to j do
6         r := r + 1
7     j := j/2
8 return r
```

7. (9 pts) Show that, if c is a positive real number, then $g(n) = 1 + c + c^2 + \cdots + c^n$ is

   a) $\Theta(1)$ if $c < 1$

   b) $\Theta(n)$ if $c = 1$

   c) $\Theta(c^n)$ if $c > 1$