COMPSCI 311: Introduction to Algorithms **Fall 2022**

Discussion 2

Your Name: ____________________ Collaborators: ____

You will be randomly assigned groups to complete these problems. List your group members on your worksheet and submit to Gradescope at the end of class.

Problem 1. Asymptotics: K&T Ch. 2 Ex. 5 Assume you have functions f and g such that $f(n)$ is $O(g(n))$. For each of the following statements, decide whether you think it is true or false and give a proof or counterexample.

- $f(n)^2$ is $O(g(n)^2)$
- $2^{f(n)}$ is $O(2^{g(n)})$

Problem 2. Asymptotics. Take the following list of functions and arrange them in ascending order of growth rate. That is, if function $g(n)$ immediately follows function $f(n)$ in your list then it should be the case that $f(n)$ is $\mathcal{O}(g(n))$.

$$f_1(n) = 10^n$$
$$f_2(n) = n^{1/3}$$
$$f_3(n) = n^n$$
$$f_4(n) = \log_2 n$$
$$f_5(n) = 2^{\sqrt{\log_2 n}}$$

Problem 3. Big-O and Big-Omega For each pair of functions f and g, indicate which of the statements are true.

1. $f(n) = 2n^2$, $g(n) = n^2 + n \log n$

   (a) $f(n)$ is $O(g(n))$ (b) $f(n)$ is $\Omega(g(n))$

2. $f(n) = \sum_{i=1}^{n} i$, $g(n) = n^3$.

   (a) $f(n)$ is $O(g(n))$ (b) $f(n)$ is $\Omega(g(n))$

3. $f(n) = n^2 \log n$, $g(n) = n^2$

   (a) $f(n)$ is $O(g(n))$ (b) $f(n)$ is $\Omega(g(n))$

Problem 4. Big-O Proof. Suppose f is $O(g)$. Prove that g is $\Omega(f)$.

Problem 5. Running Time. Each of the algorithms below prints some number of “X” characters and “Y” characters. For each algorithm, unless specific instructions are given, do the following: first write the output that would be produced with $n = 4$, then give a tight bound on its running time using Big-O notation.

1. Algorithm `Print1`(n)

```
for $i$=1 to $n$ do
    print “X”
for $j$=1 to $n$ do
    print “Y”
```

2. Algorithm `Print2`(n)

```
for $i$=1 to $n$ do
    print “X”
    if $i == 1$ then
        for $j$=1 to $n$ do
            print “Y”
```

3. Algorithm `Print3`(n)

```
for $i$=1 to $n$ do
    print “X”
    if $i \leq n/2$ then
        for $j$=1 to $n$ do
            print “Y”
```

4. A poem has m words printed on n lines. There may be any number of blank lines. Consider the following algorithm:

   Algorithm `PrintPoem`(...)

```
for each line of the poem do
    print “X”
    for each word on the line do
        print “Y”
```

   - How many times is “X” printed?
   - How many times is “Y” printed?
   - Give a Big-O running time bound for the algorithm in terms of m and n.
   - Now assume there are no blank lines. Can you give a Big-O bound that depends on only one of m or n?

5. Algorithm `Print4`(n)

```
while $n > 1$ do
    print “X”
    Set $n = n/2$
```

   (Use $n = 16$ instead of $n = 4$ for this problem)