CS 320, Fall 2021 WA1: Complexity, Orders of Magnitude

Problem Set	Topic	Max Pts	Your Score
I	Comparing functions	20	
II	Function Clubs	30	
III	Analyzing Programs	50	
Total		100	

Study the lectures on Big Oh and read Chapter 3 of the CLRS text.

ProblemSet I: Growth Rates of Functions [20 pts]

In each question below, circle one of the choices, Yes or No.

1. $f(n) = n^2 + 2n + 1000$.

   Is $f(n)$ $O(n)$? Yes / No [2 pts]

   Is $f(n)$ $O(n^2)$? Yes / No [2 pts]

2. $f(n) = 3n^2 \log^2 n + 10^6$.

   Is $f(n) \ll n^2$? Yes / No [2 pts]

   Is $f(n)$ $\Theta(n^2 \log n)$? Yes / No [2 pts]

3. $f(n) = 3^n + 2n^5 + n^3$.

   Is $f(n)$ $O(n^3)$? Yes / No [2 pts]

   Is $f(n)$ $O(2^n)$? Yes / No [2 pts]

4. $f(n) = 3^n$, $g(n) = 2^n + n^2$.

   Is $f(n) \ll g(n)$? Yes / No [2 pts]

   Is $g(n) \ll f(n)$? Yes / No [2 pts]

5. $f(n) = \log_3 n$, $g(n) = \log_8 n$.

   Is $f(n)$ $O(g(n))$? Yes / No [2 pts]

   Is $g(n)$ $O(f(n))$? Yes / No [2 pts]

ProblemSet II: The Clubs of Functions [30 pts]

In the following problems, we say that a function $f(n)$ ***swoops up*** if it grows faster than any straight line, however high its slope. We say that it ***swoops right*** if any straight line, of arbitrarily small slope grows faster than $f(n)$. A function $f(n)$, ***slowed down*** by a factor of $g(n)$ refers to the product of the two, i.e., the function $f(n)g(n)$.

Classify each of the following functions into their respective clubs: E (exponential), P (polynomial) and PL (poly-logarithmic) or neither

1. $f_1(n) = n^2 + 2n + 1000$ E / P / PL / neither [1 pt]
2. $f_2(n) = 3n^2 \log^2 n + 10^6$ E / P / PL / neither [1 pt]
3. $f_3(n) = 3^n + 2n^5 + n^3$ E / P / PL / neither [1 pt]
4. $f_4(n) = 3^n n^2$ E / P / PL / neither [1 pt]
5. $f_5(n) = n^2 \lg^2 n$ E / P / PL / neither [1 pt]
6. $f_6(n) = 5 \log_8^3 n$ E / P / PL / neither [1 pt]

In each of the True/False questions below, the correct answer is worth 1 point, and the justification is worth 3 points.

1. $f_1(n)$ swoops up on a semi-log plot T / F [4 pts]

2. $f_2(n)$ is indistinguishable from a straight line on a log-log plot T / F [4 pts]

3. $f_2(n)$ is a straight line on a log-log plot T / F [4 pts]

4. $f_3(n)$ swoops right on a log-log plot T / F [4 pts]

5. $f_4(n)$ swoops up on a log-log plot T / F [4 pts]

6. $f_6(n)$ swoops right on a log-log plot T / F [4 pts]

ProblemSet III: Analyzing the complexity of program fragments [50 pts]

For each of the code fragments below, determine the best (lowest) worst case big-Oh complexity as a function of n. Please show the complete derivation of your answers. Use additional sheets of paper as needed.

1.
```
def f1(n):
    c = 0
    step = n
    while (step > 0):
        step /= 3  # assume integer division
        for i in range(n):
          c++
    return c
```

2.
```
def f2(n):
    c = 0
    step = 1
    while step < n:
        step *= 3
        i = 1
        while i<n:
            c++
            i = 2*i
    return c
```

3.
```
def f3(n):
    c = 0
    for i in range(n):
        for j in range(i):
            step = 1
            while step < n:
                step *= 3
                c++
    return c
```

4.
```
def f4(n):
    if n<1:
        return 1
    return f4(n-1) + f4(n-1)
```

5.
```
def f5(n):
    c = 0
    for i in range (n):
      j = i
      while j>0 :
          c++
          j = 2*j - n
    return c
```