Homework 4 - Analysis

In this homework, we are going to work to become comfortable with the mathematical notation used in algorithmic analysis.

Problem 1: Quantifiers

For each of the following, write an equivalent *English statement.* Then decide whether those statements are true if x and y are integers (e.g., they can be any integer). Then write a convincing argument to prove your claim.

What's a convincing argument? It's a series of statements that can be a mathematical proof. It should be a series of connected statements that would help you explain these answers to a student in your class who was confused. These statements should be accompanied by an English language explanation, and not be solely in mathematical notation.

1. $\forall x \, \exists y \, : \, x + y = 0$

2. $\exists y \, \forall x : \, x + y = x$

3. $\exists x \, \forall y : \, x + y = x$

Name: _____ Homework 4 - Analysis

Problem 2: Growth of Functions

Organize the following functions into six (6) columns. Items in the same column should have the same asymptotic growth rates (they are big-O and big-θ of each other). If a column is to the left of another column, all of its growth rates should be slower than those of the column to its right.

$$n^2,\ n!, n\,log_2\,n, 3n, 5n^2 + 5n, 2^n, 10000, n\,log_3\,n, 100, 100n$$

Problem 3: Function Growth Language

Match the following English explanations to the *best* corresponding big-O function by drawing a line from an element in the left column to an element in the right column.

Constant	$O(n^3)$
Logarithmic	$O(1)$
Linear	$O(n)$
Quadratic	$O(log_2\,n)$
Cubic	$O(n^2)$
Exponential	$O(n!)$
Factorial	$O(2^n)$

Name: _____ Homework 4 - Analysis

Problem 4: Summation Simplification

When we are counting operations from loops, we will often need to use sigma notation to describe the summation of all of the operations in a loop. We'll need to be able to work with summations and simplify them to basic algebraic expressions when we want to count exactly the number of operations in an algorithm as described above.

Review the proof by induction that $\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$.

1. Write the following mathematical sums using sigma notation.
   a. Write the sigma expression for $1^2 + 2^2 + 3^2 + 4^2 + \cdots + n^2$

   b. Write the sigma expression for $1 + 3 + 5 + \cdots + (2n - 1)$ Remember that you can express some integers as $2k$ and others as $2k - 1$.

2. Given the expression $1^2 + 2^2 + 3^2 + 4^2 + \cdots + n^2 = \frac{1}{6}n(n+1)(2n+1)$, prove that this is true for all integers $n \geq 1$

3. Simplify the following summation: $\sum_{k=2}^{n}(k-1)$. Show all the steps of simplification and include English explanations for steps. Explain this process step-by-step the way you would to a new CS5002 student who has never simplified a summation before, or who is only familiar with the simplification $\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$

Name: _____ Homework 4 - Analysis

Problem 5: Big-O

Use the video Asymptotic Notation Part 1 for a review of the formal definition of big O. You can also find a definition of big O on page 47 (Chapter 3) of CLRS, if you would like a more theoretical explanation of the big O asymptotic upper bound.

For these problems, please show every step of your calculations, and include English statements to clarify what operations you are using to get to the next step.

Formal definition:

f(n) is O(g(n)) if there exists constants c and k such that f(n) <= c * g(n) where n > k

1. Using the definition of big-O, show that $200n + 5 \in O(2n)$

2. Using the definition of big-O, show that $n^3 + n^2 + 22 \in O(n^3)$

1. Using the definition of big-O, show that $n^{39} + 1{,}000{,}000 \in O(n^{39})$

Name: _____ Homework 4 - Analysis

Problem 6 – Insertion Sort

1. Explain what you think the worst case, big O complexity and the best-case, big O complexity of insertion sort is as you've implemented it for HW4. Make an argument for what the big O of your implementation of insertion sort is, based on your code.