CS5008 Saleh Alkhalifa
Assignment 9 4/4/2022

Assignment 9 – Primer on Proofs

Problem 1 – Quantifiers

Write the following statements as English sentences, then decide whether those statements are true if x and y can be any integers. When deciding if x and y can be any integers, prove your claim with a convincing argument.

1. **∀x∃y : x + y = 0**

   English: *"For all elements x there exists an element y, such that x plus y is equal to zero."*

   <u>Proof by cases:</u>

   Let us assume that x + y = 0, and therefore any element -x + y = 0:

Case 1: x < 0	Case 2: x > 0	Case 3: x = 0
x = -1	x = 1	x = 0
y = -x = 1	y = -x = -1	y = x = 0
∴ -1 + 1 = 0	∴ 1 – 1 = 0	∴ 0 + 0 = 0

   Therefore, it is **true**.

2. **∃y∀x : x + y = x**

   English: *"There is an element y for all elements x, such that x plus y is equal to x."*

   Proof (Definition):

   The sum of any number and 0, is equal to that number, such that

$$x + 0 = x \quad \rightarrow \quad y = 0$$

   … which is true for all elements x given the identity property.

   Therefore, it is **true**.

3. **∃x∀y : x + y = x**

$x = 1, y = 1$	$x = 2, y = 2$	$x = 1, y = 2$
$x + y = x$ $1 + 1 \neq 1$ $\therefore false$	$x + y = x$ $2 + 2 \neq 2$ $\therefore false$	$x + y = x$ $1 + 2 \neq 1$ $\therefore false$

   Therefore, it is **false**.

Assignment 9 – Primer on Proofs

Problem 2 – Growth of Functions

Organize the following functions into six 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 its growth rates should be slower than those of the column to its right.

O(1)	**O(Log n)**	**O(n)**	**O(n log n)**	**O(n^2)**	**O(n^3)**	**O(2^N)**	**O(N!)**
10000 100		3n	n $\log_2$ n n $\log_3$ n	n^2 $5n^2+3$		2^n	N!

O(1)	**O(n)**	**O(n log n)**	**O(n^2)**	**O(2^N)**	**O(N!)**
10000 100	3n 100n	n $\log_2$ n n $\log_3$ n	n^2 $5n^2+3$	2^n	n!

Problem 3 – Function Growth Language

Match the following English explanations to the best corresponding Big-O function by drawing a line from the left to the right.

1	Constant Time	O(1)
2	Logarithmic Time	O(log n)
3	Linear Time	O(n)
4	Quadratic Time	O(n^2)
5	Cubic Time	O(n^3)
6	Exponential Time	O(2^n)
7	Factorial Time	O(n!)

Problem 4 – Big-O

1. **Using the definition of big-O, show 100n + 5 = O(2n).**

$$100n + 5 \ \le c(2n)$$
$$-100n\ 100n + 5 \ \le c(2n) - 100n$$
$$5 \le c(2n) - 100n$$
$$5 \le 2cn - 100n$$
$$5 \le 2n - (c - 50)$$
$$\div\ 2n\ 5 \le 2n - (c - 50) \div 2n$$
$$\frac{5}{2n} \le (c - 50)$$
$$50 + \frac{5}{2n} \le (c - 50) + 50$$
$$\frac{5}{2n} + 50 \le c$$

When n=1, k=1, using our equation

$$\frac{5}{2n} + 50 \le c$$
$$\frac{5}{2 * 1} + 50 \le c$$
$$c \ge \frac{105}{2}$$

∴ The function is O(2n) because when the constant k=1 and constant c = 105/2, the definition:

$$O \ \le 100n + 5 \ \le O(2n)$$
$$O \le f(n) \le c * g(n)$$

... for all positive numbers greater than or equal to zero.

Saleh Alkhalifa
4/4/2022

Assignment 9 – Primer on Proofs

2. **Using the definition of big-O, show $n^3 + n^2 + n + 100 = O(n^3)$.**

$$n^3 + n^2 + n \leq 2n^3$$

$$n^3 + n^2 + n + 200 \leq 2n^3 + 200n^3$$

$$n^3 + n^2 + n + 200 \leq 102(n^3)$$

$$n^3 + n^2 + n + 200 \leq \mathbf{102}(n^3)$$

$$n^3 + n^2 + n + 200 = O(n^3)$$

$$\therefore Complexity = O(n^3)$$

3. **Using the definition of big-O, show $n^{99} + 10000000 = O(n^{99})$).**

$$n^{99} + 10{,}000{,}000 \ = O(n^{99})$$

$$n^{99} + 10{,}000{,}000 \ \leq 10{,}000{,}000 * n^{99} + n^{99}$$

$$n^{99} + 10{,}000{,}000 \ \leq 10{,}000{,}001 * n^{99}$$

$$n^{99} + 10{,}000{,}000 \ \leq \mathbf{10{,}000{,}001} * n^{99}$$

$$n^{99} + 10{,}000{,}000 \ \leq \mathrm{O}(n^{99})$$

$$\therefore Complexity = O(n^{99})$$

Problem 5 – Searching

We will consider the problem of search in ordered and unordered arrays.

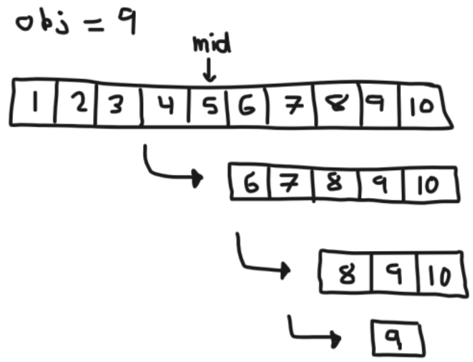
1. **We are given an algorithm called search which can tell us true or false in one step per search query if we have found our desired element in an unordered array of length 2048. How many steps does it take in the worse possible case to search for a given element in the unordered array?**

   To search for a given element in the unordered array, as a worst-case scenario it would take 2048 steps using the T or F methodology listed, since it is possible that the desired element could be the last one in the array.

Assignment 9 – Primer on Proofs

2. **Describe a fasterSearch algorithm to search for an element in an ordered array. In your explanation, include the time complexity using Big-O notation and draw or otherwise explain clearly why this algorithm is able to run faster.**

   A faster searching method for an ordered array could be done using binary search, in which half of the array is eliminated in each step leading to a much more favorable complexity. The idea here is that given a number of interest, we can find the midpoint along a list and eliminate the half that is not of interest to us, and continue this process until the desired value is found. For example, if the objective value is 9, then the list could be looked it, establishing the midpoint, and eliminating the list iteratively until the value of interest is found:



3. **How many steps does your fasterSearch algorithm (from the previous part) take to find an element in an ordered array of length 256 in the worse-case? Show the math to support your claim**

   When it comes to Binary search, the complexity for this algorithm is O(log n), and we can use this complexity to show the number of steps it would take to find an element of interest in a given ordered array:

$$log_2 256 = n$$

$$2^n = 256$$

$$n = 8$$

$$\therefore It\ would\ take\ 8\ steps\ to\ find\ the\ element\ of\ interest$$

CS5008 | Saleh Alkhalifa
Assignment 9 | 4/4/2022

Assignment 9 – Primer on Proofs

Problem 6 – Searching

1. **Describe an algorithm for finding the real coin. You must also include the algorithm the time complexity. *Hint* Think carefully–or do this experiment with a roommate and think about how many ways you can prune the maximum amount of fake coins using your scale.**

   In the field of Analytical Chemistry (my undergraduate degree), we often face this exact problem when it comes to elements on the periodic table. In some cases, elements can have isotopes where the weight of 1 atom for example can be 1.00000 g/mol, and its isotope 1.00001 g/mol, and distinguishing between is often the objective.

   We can use a method similar to binary search to help us identify the atom, or coin of interest from its counter parts. First, you can add 50 coins to each side of the scale. The side with the fake coin will present itself since the scale will be tipped, allowing us to eliminate half the coins in the first iteration. We can repeat this process over and over until the problematic coin is found. Identifying the count may happen in one of two ways:

   a. You may run out of coins through elimination and find the coin of interest
   b. The balance may not tip, and therefore the coin of interest was removed to keep the piles the same (100 -> 50 -> 25 -> 12 – we will need to remove one here)

2. **How many weighing must you do to find the real coin given your algorithm?**

$$log_2 100 = n$$

$$2^n = 100$$

$$n = 6.6$$

$$n = 6$$

$$\therefore It\ would\ likely\ take\ 6\ steps\ to\ find\ the\ element\ of\ interest$$