CSCE 222 [Sections 503, 504] Discrete Structures for Computing
Fall 2019 – Hyunyoung Lee

Problem Set 4

Due dates: Electronic submission of *yourLastName-yourFirstName-hw4.tex* and *yourLastName-yourFirstName-hw4.pdf* files of this homework is due on **Monday, 9/30/2019 before 10:00 p.m.** on `http://ecampus.tamu.edu`. You will see two separate links to turn in the .tex file and the .pdf file separately. Please do not archive or compress the files. **If any of the two submissions are missing, you will likely receive zero points for this homework.**

Name: Andrew Han **UIN: 227009495**

Resources. Dr.Lee's slides on Sets and Functions, Complexity of Algorithims

On my honor, as an Aggie, I have neither given nor received any unauthorized aid on any portion of the academic work included in this assignment. Furthermore, I have disclosed all resources (people, books, web sites, etc.) that have been used to prepare this homework.

Electronic Signature: **Andrew Han**

Total 100 points.

Problem 1. (5 points) Let x be a real number and n an integer. Show that

$$\lceil x \rceil = n \text{ if and only if } x \leqslant n < x + 1.$$

[Hint: Use the definition for $\lceil x \rceil = n$ given as the second fact among the four facts in slide #39 in the lecture slides on Sets and Functions. What you are proving here is the last fact in the same slide.]

Solution. One of the facts on slide #39 is that

$$\lceil x \rceil = n \text{ if and only if } n - 1 < x \leqslant n.$$

From this, you can derive a new inequality by looking at the separate parts. We can take the right side of the inequality and have n be in the middle.

$$x \leqslant n$$

Then we can rearrange the left part of the inequality to be

$$n < x + 1$$

Putting these together, we get

$$x \leqslant n < x + 1$$

Problem 2. (15 points) Let n be a positive integer. Show that if n is a perfect square, then

$$\lfloor\sqrt{n}\rfloor - \lfloor\sqrt{n-1}\rfloor = 1.$$

[Hint: Use the definition for $\lfloor x\rfloor = n$ given as the first fact among the four facts in slide #39 in the lecture slides on Sets and Functions.]

Solution. The definition of a floor is

$$\lfloor x\rfloor = n \text{ if and only if } n \leqslant x < n+1$$

We can substitute the x value as n in the case, and replace n with x to get

$$x \leqslant \sqrt{n} < x+1$$

If n is a perfect square, $\sqrt{n}$ will be an integer by the definition of a perfect square. If the result of the square root is an integer,

$$\lfloor\sqrt{n}\rfloor \text{ will be equal to } \sqrt{n}$$

Looking at $\sqrt{n-1}$, we just stated that $\sqrt{n}$ is going to be an integer because it is a perfect square. This means that $\sqrt{n-1}$ will return a value that is not an integer and will be less than $\sqrt{n}$. This means that the floor of $\sqrt{n-1}$ will have to be the next lowest integer from $\sqrt{n}$. Therefore,

$$\lfloor\sqrt{n}\rfloor - \lfloor\sqrt{n-1}\rfloor = 1$$

Problem 3. (10 points) Let f_1, f_2, f_3 be functions from the set $\mathbf{N}$ of natural numbers to the set $\mathbf{R}$ of real numbers. Suppose that $f_1 = O(f_2)$ and $f_2 = O(f_3)$. Is it possible that

$$f_1(n) > f_3(n)$$

holds for all natural numbers n? Give an example for f_1, f_2, and f_3 if it is possible, or give an explanation (convincing argument or proof) if it is impossible. [Hint: Think carefully about the *definition* of a function being in big-Oh of another function *as given in class*.]

Solution. In order for a function to be in the big-oh of another funtction, the following needs to be true

$$|f(n)| \leqslant U|g(n)| \text{ for all } n \geqslant n_0$$

It is given that $f_1 = O(f_2)$. This means that

$$|f_1(n_1)| \leqslant U_1|f_2(n_1)| \text{ for all } n_1 \geqslant n_0$$

It is also given that $f_2 = O(f_3)$. This means that

$$|f_2(n_2)| \leqslant U_2|f_3(n_2)| \text{ for all } n_2 \geqslant n_0$$

With the given statements about f_1, f_2, f_3 and which functions are in the big-oh function of others, we know that f_1 is upper bounded by f_2 and f_2 is upper bounded by f_3. Therefore, it is not possible that $f_1(n) > f_3(n)$

Problem 4. (10 pts $\times$ 3 = 30 points) Determine whether each of the following statements is true or false. In each case, answer true or false, and justify your answer (by giving a direct proof if it is true, or a proof by contradiction if it is false; always use the definition involving the absolute values, as given in class).

a) $3n^2 + 41 = O(n^3)$

Solution. True. In order for the statement above to be true,

$$|3n^2 + 41| \leqslant U|(n^3)| \text{ for all n } \geqslant n_0$$

Because n is a positive integer, we can get rid of the absolute value bars and simplify the expression to

$$\frac{3n^2 + 41}{n^3} \leqslant U$$

If we set n_0 to 10 and U to .341, the expression will hold for all $n \geqslant n_0$ making the claim hold true.

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

Solution. False. In order for a function to in the big-oh of another, the inequality stated above needs to be satisfied. If we simplify this expression above we get

$$\frac{n^3 + 2n + 3}{n^2} \leqslant U$$

The expression will continue to grow as n increases because n^3 is in the numerator and n^2 is the denominator. This means that there will be no constant we can set so that the function will not outgrow it, making the claim false.

c) $\frac{1}{2}n^2 + 5 = \Omega(n)$

Solution. True. The definition for big-omega in this case would be

$$L|n| \leqslant |\frac{1}{2}n^2 + 5| \text{ for all n } \geqslant n_0$$

Dividing both sides by n and dropping the absolute value bars yields

$$L \leqslant |\frac{1}{2}n + \frac{5}{n}| \text{ for all n } \geqslant n_0$$

This will be a growing function as n increases. If n_0 is set to 20 and L is set to 10.25, the expression will hold for all $n \geqslant n_0$ making the claim true.

Problem 5. (10 points) Let k be a fixed positive integer. Show that

$$1^k + 2^k + \cdots + n^k = O(n^{k+1})$$

holds.

Solution. In order for the function on the left to be in big-oh of $n^k + 1$, we need to show that it will always be upper bounded by $n^k + 1$. Using the definition of big-oh, the expression can be rearranged to

$$\frac{1^k + 2^k + ... + n^k}{n^k + 1} = U$$

The denominator of the fraction will always have an exponent that is one degree higher than the elements being added in the numerator. This means that each element will be being divided by a bigger number. The claim will hold true for U that is determined based off of what n and k are set to.

Problem 6. (15 points) Let f_1, f_2, f_3, f_4 be functions from the set $\mathbf{N}$ of natural numbers to the set $\mathbf{R}$ of real numbers. Suppose that $f_1 = O(f_2)$ and $f_3 = O(f_4)$. Use the *definition* of Big Oh *given in class* to prove that

$$f_1(n) + f_3(n) = O(\max(|f_2(n)|, |f_4(n)|)).$$

Solution. The definition of big-oh for this problem would be

$$|f_1(n)| + |f_3(n)| \leqslant U|(\max(|f_2(n)|, |f_4(n)|))|$$

It is given that $f_1 = O(f_2)$ and $f_3 = O(f_{4)}$. This means that if f_2 is larger than f_4, f_1 will be upper bounded by f_2 (by what is given) and f_3 will also be upper bounded by f_2 due to the fact that $f_3 = O(f_4)$ and $f_2 > f_4$. This will also apply for when f_4 is larger than f_2. The claim above will hold true if both f_1 and f_3 are upper bounded by $\max(|f_2(n)|, |f_4(n)|)$.

Problem 7. (5 pts $\times$ 3 = 15 points) Suppose that you have two algorithms A and B that solve the same problem. Algorithm A has worst case running time $T_A(n) = 2n^2 - 2n + 1$ and Algorithm B has worst case running time $T_B(n) = n^2 + n - 1$.

a) Show that both $T_A(n)$ and $T_B(n)$ are in $O(n^2)$.

b) Show that $T_A(n) = 2n^2 + O(n)$ and $T_B(n) = n^2 + O(n)$.

c) Explain which algorithm is preferable.

Solution. a) For $T_A(n)$ to be in $O(n^2)$, the following needs to be true.

$$|2n^2 - 2n + 1| \leqslant U|n^2|$$

Dividing both sides by n^2 and dropping the absolute value bars gives us

$$2 - \frac{2}{n} + \frac{1}{n^2} \leqslant U \text{ for all n } \geqslant n_0$$

If n_0 is set to 1 and U is set to 2, the claim will hold true for algorithm A because $\frac{2}{n}$ will always subtract more than $\frac{1}{n^2}$ can add.

For For $T_B(n)$ to be in $O(n^2)$, the following needs to be true.

$$|n^2 + n - 1| \leqslant U|n^2|$$

Dividing both sides by n^2 and dropping the absolute value bars gives us

$$1 + \frac{1}{n} - \frac{1}{n^2} \leqslant U \text{ for all n } \geqslant n_0$$

If n_0 is set to 2 and U is set to 1.25, the claim will hold true for algorithm B.
b) Subtract $T_A(n)$ by $2n^2$ to get

$$-2n + 1 = O(n)$$

By the defintion of big-oh notation

$$|-2n+1| \leqslant U|n| \text{ for all n } \geqslant n_0$$

$$|-2+\frac{1}{n}| \leqslant U \text{ for all n } \geqslant n_0$$

If n_0 is set to 1 and U is 2, the claim will hold true for $T_A(n)$.

Subtract $T_B(n)$ by n^2 to get

$$n - 1 = O(n)$$

By the definition of big-oh notation

$$|n-1| \leqslant U|n| \text{ for all n } \geqslant n_0$$

$$|1-\frac{1}{n}| \leqslant U \text{ for all n } \geqslant n_0$$

If n_0 is set to 1 and U is 1, the claim will hold true for $T_B(n)$. This means that algorithm B has a shorter worst-case run time, making it the preferable algorithm to use.

c) In part B, we proved that $T_A(n) = 2n^2 + O(n)$ and $T_B(n) = n^2 + O(n)$

Checklist:
☐ Did you type in your name and UIN?
☐ Did you disclose all resources that you have used?
(This includes all people, books, websites, etc. that you have consulted.)
☐ Did you electronically sign that you followed the Aggie Honor Code?
☐ Did you solve all problems?
☐ Did you submit both of the .tex and .pdf files of your homework to the correct link on eCampus?