CSCE 222 [Sections 502, 503] Discrete Structures for Computing
Spring 2017 – 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 **Friday, 2/24/2017 before 11:00 a.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. A signed paper copy of the pdf file is due on **Friday, 2/24/2017** *at the beginning of class.* **If any of the three submissions are missing, your work will not be graded.**

Name: Joseph Martinsen **Section: 503**

Resources. (All people, books, articles, web pages, etc. that have been consulted when producing your answers to this homework.)

```
https://www.youtube.com/watch?v=P2qHss2-aSQ
https://www.youtube.com/watch?annotation_id=annotation_2862598731&feature=iv&
  index=3&list=PLj68PAxAKGowkG1QYgun4DrByPwsyB04h&src_vid=P2qHss2-aSQ&v=DjfYhHSkWqo
https://www.youtube.com/watch?annotation_id=annotation_4083618415&feature=iv&src_vid=P2
  Vzqaz4MDGvc
```

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.

Signature: ____

Problem 1. (10 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. By definition of Big Oh,
If $f_1 \in O(f_2)$, then there exist a $k_1 \in \mathbb{R}$ and $n_1 \in \mathbb{N}$ such that $f_1(n) \leq Af_2(n)$ for all $n > n_1$
In the same manner, if $f_3 \in O(f_4)$, then there exist a $k_2 \in \mathbb{R}$ and $n_2 \in \mathbb{N}$ such that $f_1(n) \leq Af_2(n)$ for all $n > n_2$
Let $N = \max(n_1, n_2)$. For all $n > N$ it is true that

$$f_1(n) + f_3(n) \leq k_1 f_2(n) + k_2 f_4(n)$$

Let $K = \max(k_1, k_2)$ and $f_5(n) = \max(f_2(n), f_4(n))$

$$\begin{aligned} f_1(n) + f_3(n) &\leq K[f_2(n) + f_4(n)] \\ &\leq K f_5(n) = K \max(f_2(n), f_4(n)) \\ f_1(n) + f_3(n) &\leq K \max(f_2(n), f_4(n)) \end{aligned}$$

By definition of Big Oh, for $n > N$

$$f_1(n) + f_3(n) \in O(\max(f_2(n), f_4(n))) \quad \square$$

Problem 2. (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 or given an argument that this is impossible.

Solution. Allow $f_3 = f_2 + 1$ and $f_2 = f_1 + 1$, where $f_1 = O(f_2)$ and $f_2 = O(f_3)$ is still true.

$$\begin{aligned} f_3 &= f_2 + 1 \\ f_3 &= f_1 + 1 + 1 \\ f_3 &= f_1 + 2 \end{aligned}$$

For all n, f_3 is 2 bigger than f_1. It then follows that $f_1 < f_3 \forall n \quad \square$

Problem 3. (5 pts $\times$ 4 = 20 points) Determine whether each of the following statements is true or false. In each case, answer true or false, and justify your answer.

a) $3n^2 - 42 = O(n^2)$

$$\begin{aligned} 3n^2 - 42 &\leq Un^2 \\ &\leq 3n^2 + n^2 \\ &\leq 4n^2 && \text{for n} > 1 \end{aligned}$$

$\therefore 3n^2 - 42$ is $O(n^2)$ with $U = 4$ and $n > 1$ as witnesses $\quad \square$

b) $n^2 = O(n \log n)$

$$\frac{n^2}{n \log n} = \frac{n}{\log n}$$

As n grows, this does not approach nor is there C that satisfies this equation for all n $\therefore n^2$ is NOT $O(n \log n)$

c) $1/n = O(1)$

$$\begin{aligned} \frac{1}{n} &\leq U1 \\ \frac{1}{n} &\leq 1 && \text{for n} > 1 \end{aligned}$$

$\therefore \frac{1}{n}$ is $O(1)$ with $U = 1$ and $n > 1$ as witnesses $\square$

d) $n^n = \Omega(2^n)$

Solution.

Problem 4. (15 points) Does $\Theta(n^3 + 2n + 1) = \Theta(n^3)$ hold? Justify your answer.

Solution.
A function $f(n)$ is $\Theta(g(n))$ iff
$f(n) \leq Ug(n)$ for all $n > n_0$ O *definition*
and
$f(n) \geq Lg(n)$ for all $n > n_0$ Ω *definition*
First, check if $O(n^3 + 2n + 1) = O(n^3)$ holds

$$\begin{aligned} n^3 + 2n + 1 &\leq Un^3 \\ &\leq n^3 + 2n^3 + n^3 \\ &\leq 4n^3 \end{aligned}$$

$\therefore O(n^3 + 2n + 1) = O(n^3)$ with $U = 4$ and $n \geq 1$ as witnesses
Next, check $\Omega(n^3 + 2n + 1) = \Omega(n^3)$

$$\begin{aligned} n^3 + 2n + 1 &\geq Ln^3 \\ n^3 + 2n + 1 &\geq n^3 && \text{for } n \geq 1 \\ 1 + \frac{2}{n^2} + \frac{1}{n^3} &\geq 1 \\ 1 + \frac{2}{1} + \frac{1}{1} &\geq 1 && \text{for } n \geq 1 \end{aligned}$$

$\therefore O(n^3 + 2n + 1) = O(n^3)$
$\therefore \Omega(n^3 + 2n + 1) = \Omega(n^3)$
$\therefore \Theta(n^3 + 2n + 1) = \Theta(n^3)$ $\square$

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.

$$\begin{aligned} 1^k + 2^k + \cdots + n^k &\le Un^{k+1} \\ &\le n^k + n^k + \cdots + n^k \\ &\le n \cdot n^k \\ &\le 1 \cdot n^{k+1} & \text{for n} > 1 \end{aligned}$$

$\therefore 1^k + 2^k + \cdots + n^k$ is $O(n^{k+1})$ with $U = 1$ and $n > 1$ as witnesses $\square$

Problem 6. (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 $T_B(n) = n^2 + n - 1$.

Solution.

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

$$\begin{aligned} T_A(n) = 2n^2 - 2n + 1 &\le Un^2 \\ &\le 2n^2 + n^2 + n^2 \\ &\le 4n^2 & \text{for } n > 1 \end{aligned}$$

$\therefore T_A(n)$ is $O(n^2)$ with $U = 4$ and $n > 1$ as witnesses

$$\begin{aligned} T_B(n) = n^2 + n - 1 &\le Un^2 \\ &\le 2n^2 + n^2 + n^2 \\ &\le 4n^2 & \text{for } n > 1 \end{aligned}$$

$\therefore T_B(n)$ is $O(n^2)$ with $U = 4$ and $n > 1$ as witnesses

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

$$\begin{aligned} -2n + 1 &\le Un \\ &\le n + n \\ &\le 2n \\ -2n + 1 &\in O(n) \end{aligned}$$

$\therefore T_A(n) = 2n^2 - 2n + 1 = 2n^2 + O(n)$

$$\begin{aligned} n - 1 &\le Un \\ &\le n + n \\ &\le 2n \\ n - 1 &\in O(n) \end{aligned}$$

$\therefore T_A(n) = n^2 + n - 1 = n^2 + O(n)$

c) Explain which algorithm is preferable.
For large n $T_B(n)$ is preferable because due to the fact that the coefficient of n^2 is 1 instead of 2, like in $T_A(n)$

Problem 7. (10 points) Section 3.3, Exercise 14 on page 230.

Solution.

a)
$n = 2$ $a_2 = 3$ $a_1 = 1$ $a_0 = 1$
Line 1: $y := 3$
For Loop:
$i := 1 \quad y := 3 * 2 + 1 => 7$
$i := 2 \quad y := 7 * 2 + 1 => 15$
return 15

b)
There is one multiplication and one addition within the for loop. The loop executes n times for $x = n$.
$\therefore$ there are n additions and n multiplications

Problem 8. (10 points) Section 3.3, Exercise 16 a), d), g) and h) on page 230.

Solution.
Seconds in a day = $10^5 sec$
\# of Operations = 10^{16}
a)

$$\begin{aligned} f(n) &\leq 10^{16} \\ \log n &\leq 10^{16} \\ n &\leq 2^{10^{16}} \end{aligned}$$

Largest value of n is $2^{10^{16}}$
d)

$$\begin{aligned} f(n) &\leq 10^{16} \\ 1000n^2 n &\leq 10^{16} \\ n^2 &\leq 10^{13} \\ n &\leq 10^{13/2} \end{aligned}$$

Largest value of n is $10^{13/2}$
g)

$$
\begin{aligned}
f(n) &\leq 10^{16} \\
2^{2n} &\leq 10^{16} \\
2n &\leq \log_2 10^{16} \\
n &\leq \frac{1}{2}\log_2 10^{16}
\end{aligned}
$$

Largest value of n is $\frac{1}{2}\log_2 10^{16}$

h)

$$
\begin{aligned}
f(n) &\leq 10^{16} \\
2^{2^2 n} &\leq 10^{16} \\
2^n &\leq \log_2 10^{16} \\
n &\leq \log_2(\log_2 10^{16})
\end{aligned}
$$

Largest value of n is $\log_2(\log_2 10^{16})$

Checklist:

- ☐ Did you type in your name and section?
- ☐ Did you disclose all resources that you have used?
  (This includes all people, books, websites, etc. that you have consulted.)
- ☐ Did you sign that you followed the Aggie Honor Code?
- ☐ Did you solve all problems?
- ☐ Did you submit the .tex and .pdf files of your homework to the correct link on eCampus?
- ☐ Did you submit a signed hardcopy of the pdf file in class?