CSCE 222 Discrete Structures for Computing – Fall 2023
Hyunyoung Lee

Problem Set 5

Due dates: Electronic submission of *yourLastName-yourFirstName-hw5.tex* and *yourLastName-yourFirstName-hw5.pdf* files of this homework is due on **Monday, 10/23/2023 before 11:59 p.m.** on `https://canvas.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 files are missing, you will receive zero points for this homework.**

Name: Kevin Lei **UIN: 432009232**

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

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 answer this homework.

Electronic signature: **Kevin Lei**

Total $100 + 10$ (bonus) points.

The intended formatting is that this first page is a cover page and each problem solved on a new page. You only need to fill in your solution between the `\begin{solution}` and `\end{solution}` environment. Please do not change this overall formatting.

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 sign that you followed the Aggie Honor Code?
- ☐ Did you solve all problems?
- ☐ Did you submit both the .tex and .pdf files of your homework to each correct link on Canvas?

Problem 1. (10 points) Section 11.1, Exercise 11.3

Solution. To determine who is correct in this situation, we can use the definition of asymptotic equality. We have that $f(n) = n^2 + 2n$ and $g(n) = n^2$. Taking the limit of $\frac{f(n)}{g(n)}$ as n approaches infinity, we have:

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{n^2 + 2n}{n^2} = \lim_{n\to\infty} \frac{n+2}{n}$$

Using L'Hopital's rule, we have:

$$\lim_{n\to\infty} \frac{n+2}{n} = \lim_{n\to\infty} \frac{1}{1} = 1$$

This means that by definition, $f(n) \sim g(n)$, so Ernie is correct. Bert says that f and g cannot be asymptotically equal since $f(n) - g(n) \geq 2n$ for all $n \geq 1$. However, this is the wrong way to look at asymptotic equality. Asymptotic equality is supposed to describe the behavior of functions for large inputs. Bert is focusing on the absolute difference between the functions f and g. However, asymptotic equality deals with the relative error $(f(n) - g(n))/g(n)$. In the case of $f(n) = n^2 + 2n$ and $g(n) = n^2$, the relative error does vanish for large inputs, since it is this limit: $\lim_{n\to\infty} \frac{f(n)-g(n)}{g(n)} = \lim_{n\to\infty} \frac{2n}{n^2} = \lim_{n\to\infty} \frac{2}{n} = 0$.

Problem 2. (20 points) Section 11.3, Exercise 11.14. [Requirement: Study the definition of $\asymp$ involving the inequalities carefully and use the definition to answer the questions.]

Solution. Given that $f : \mathbf{N}_1 \to \mathbf{R}$ and $g : \mathbf{N}_1 \to \mathbf{R}$, the functions f and g have the same order of growth (represented by $f \asymp g$) if and only if there exist positive real constants c and C and a positive integer n_0 such that

$$c|g(n)| \leq |f(n)| \leq C|g(n)|$$

for all $n \geq n_0$. To show that this is an equivalence relation, we need to show the following:

(i) $f \asymp f$
Here we need to find positive real constants c and C and a positive integer n_0 such that $c|f(n)| \leq |f(n)| \leq C|f(n)|$ for all $n \geq n_0$. The obvious choice is $c = C = 1$ and $n_0 = 1$, so the relation is reflexive.

(ii) $f \asymp g$ if and only if $g \asymp f$.
Suppose that $f \asymp g$. This implies that there exist positive real constants c and C and a positive integer n_0 such that $c|g(n)| \leq |f(n)| \leq C|g(n)|$ for all $n \geq n_0$. Splitting up this inequality, we have $c|g(n)| \leq |f(n)|$ and $|f(n)| \leq C|g(n)|$. Dividing by c and C respectively, we have $|g(n)| \leq \frac{1}{c}|f(n)|$ and $\frac{1}{C}|f(n)| \leq |g(n)|$, which still holds for all $n \geq n_0$. Using this, we can write the definition of $g \asymp f$ as an inequality:

$$\frac{1}{C}|f(n)| \leq |g(n)| \leq \frac{1}{c}|f(n)|$$

Since this holds for all $n \geq n_0$, we have that $g \asymp f$, and $f \asymp g \to g \asymp f$. The same logic can be applied to show that given $g \asymp f$, $f \asymp g$ is true. Therefore, the relation is symmetric.

(iii) $f \asymp g$ and $g \asymp h$ implies $f \asymp h$.
Given that $f \asymp g$ and $g \asymp h$, we know that there exist positive real constants c_1, C_1, c_2, and C_2 and positive integers n_1 and n_2 such that $c_1|g(n)| \leq |f(n)| \leq C_1|g(n)|$ for all $n \geq n_1$ and $c_2|h(n)| \leq |g(n)| \leq C_2|h(n)|$ for all $n \geq n_2$. Using this information, we want to show that there exist positive real constants c and C and a positive integer n_0 such that $c|h(n)| \leq |f(n)| \leq C|h(n)|$ for all $n \geq n_0$. From the definitions, we have that $c_1|g(n)| \leq |f(n)|$ and $c_2|h(n)| \leq |g(n)|$. Multiplying the second inequality by c_1 and putting the two together, we have $c_1c_2|h(n)| \leq |f(n)|$. Doing the same for the upper bound, we have $|f(n)| \leq C_1C_2|h(n)|$. Now we have the inequality $c_1c_2|h(n)| \leq |f(n)| \leq C_1C_2|h(n)|$, which holds for all $n \geq max(n_1, n_2)$. This satisfies the definition of $f \asymp h$, so $f \asymp g$ and $g \asymp h$ implies $f \asymp h$.

Therefore, $\asymp$ is an equivalence relation.

Problem 3. (15 points) Prove that $3n^2 + 41 \in O(n^3)$ by giving a direct proof based on the definition of big-O involving the inequalities and absolute values, as given in the lecture notes Section 11.4.

To do so, first write out what $3n^2 + 41 \in O(n^3)$ means according to the definition. Then, you need to find a positive real constant C and a positive integer n_0 that satisfy the definition.

Solution. For functions $f : \mathbf{N}_1 \to \mathbf{R}$ and $g : \mathbf{N}_1 \to \mathbf{R}$, we say that g is an asymptotic upper bound for f, written as $f \in O(g)$, if and only if there exists a positive real constant C and a positive integer n_0 such that

$$|f(n)| \leq C|g(n)|$$

for all $n \geq n_0$. Using the definition of big-O, we have that $3n^2 + 41 \in O(n^3)$ if and only if there exist a positive real constant C and and a positive integer n_0 such that $|3n^2 + 41| \leq C|n^3|$ for all $n \geq n_0$. Since both sides of the inequality are positive for all $n \geq 1$, we can remove the absolute value signs, and we have $3n^2 + 41 \leq Cn^3$. This can be rewritten as $C \geq \frac{3n^2+41}{n^3}$. For $n = 1$, we have $C \geq \frac{3+41}{1} = 44$. For larger values of n, the fraction $\frac{3n^2+41}{n^3}$ approaches zero, so we can choose any value of C greater than 44. This means we can choose $C = 44$ and $n_0 = 1$ to satisfy the definition of big-O. Therefore, we have directly proven that $3n^2 + 41 \in O(n^3)$.

Problem 4. (15 points) Prove that $\frac{1}{2}n^2 + 5 \in \Omega(n)$ by giving a direct proof based on the definition of big-Ω involving the inequalities and absolute values, as given in the lecture notes Section 11.5.

To do so, first write out what $\frac{1}{2}n^2 + 5 \in \Omega(n)$ means according to the definition. Then, you need to find a positive real constant c and a positive integer n_0 that satisfy the definition.

Solution. For functions $f : \mathbf{N}_1 \to \mathbf{R}$ and $g : \mathbf{N}_1 \to \mathbf{R}$, we say that g is an asymptotic lower bound for f, written as $f \in \Omega(g)$, if and only if there exists a positive real constant c and a positive integer n_0 such that

$$c|g(n)| \leq |f(n)|$$

for all $n \geq n_0$. To prove that $\frac{1}{2}n^2 + 5 \in \Omega(n)$, we need to find a positive real constant c and a positive integer n_0 such that $c|n| \leq |\frac{1}{2}n^2 + 5|$ for all $n \geq n_0$. Since both functions have a domain of $\mathbf{N}_1$, we can remove the absolute value signs, and we have $cn \leq \frac{1}{2}n^2+5$. Now we can rewrite the inequality as $c \leq \frac{n}{2}+\frac{5}{n}$. Starting with $n = 1$, we have $c \leq \frac{1}{2} + 5 = \frac{11}{2}$. As n approaches infinity, the fraction $\frac{5}{n}$ approaches zero, so we can choose any value of c less than or equal to $\frac{11}{2}$. This means we can choose $c = \frac{11}{2}$ and $n_0 = 1$ to satisfy the definition of big-Ω. Therefore, we have directly proven that $\frac{1}{2}n^2 + 5 \in \Omega(n)$.

Problem 5. (10+10 = 20 points) Read Section 11.6 carefully before attempting this problem.

Analyze the running time of the following algorithm using a step count analysis as shown in the Horner scheme (Example 11.40).

```
// search a key in an array a[1..n] of length n
search(a, n, key)          cost   times
  for k in (1..n) do        c1    [ n ]
    if a[k]=key then        c2    [ n ]
       return k             c3    [ 1 ]
  endfor                    c4    [ n ]
  return false              c5    [ 1 ]
```

(a) Fill in the `[  ]`s in the above code each with a number or an expression involving `n` that expresses the step count for the line of code.

(b) Determine the worst-case complexity of this algorithm and give it in the Θ notation. Show your work and explain using the definition of Θ involving the inequalities.

Solution. (For part (b)) The time compexity of this algorithm is given by $T(n) = c_1n + c_2n + c_3 + c_4n + c_5$. Since the highest order term in this equation is n, we can say that $T(n) \in \Theta(n)$. According to the definition of Θ, this means that there exist positive real constants c and C and a positive integer n_0 such that $cn \leq T(n) \leq Cn$ for all $n \geq n_0$. Let's say that $n_0 = 1$. Starting with the lower bound, we have $cn \leq c_1n + c_2n + c_3 + c_4n + c_5$. Since n only gets larger, that means c can be any vaue less than or equal to $c_1 + c_2 + c_4$. For the upper bound, we have $c_1n + c_2n + c_3 + c_4n + c_5 \leq Cn$. Following the same logic, we can choose any value for C as long as it is greater than or equal to $c_1 + c_2 + c_4$. Since there exits positive real constants c and C and a positive integer n_0 such that $cn \leq T(n) \leq Cn$ for all $n \geq n_0$, we have shown that $T(n) \in \Theta(n)$.

Problem 6. (15+15 = 30 points) Read Section 11.6 carefully before attempting this problem. Analyze the running time of the following algorithm using a step count analysis as shown in the Horner scheme (Example 11.40).

```
// determine the number of digits of an integer n
binary_digits(n)             cost  times
  int cnt = 1                 c1    [ 1 ]
  while (n > 1) do            c2    [ log(n) ]
    cnt = cnt + 1             c3    [ log(n) ]
    n = floor( n/2.0 )        c4    [ log(n) ]
  endwhile                    c5    [ log(n) ]
  return cnt                  c6    [ 1 ]
```

(a) Fill in the `[ ]`s in the above code each with a number or an expression involving `n` that expresses the step count for the line of code.

(b) Determine the worst-case complexity of this algorithm as a function of n and give it in the Θ notation. Show your work and explain using the definition of Θ involving the inequalities.

Solution. (For part (b)) The time complexity of this algorithm is given by $T(n) = c_1 + c_2 \log(n) + c_3 \log(n) + c_4 \log(n) + c_5 \log(n) + c_6$. Since the fastest growing term in this equation is $\log(n)$, we can say that $T(n) \in \Theta(\log(n))$. Using the definition of Θ, this means that there exist constants c and C and a positive integer n_0 such that $c \log(n) \leq T(n) \leq C \log(n)$ for all $n \geq n_0$. Going with $n_0 = 1$, we can analyze the lower and upper bounds separately. Starting with the lower bound, we need to find a real constant c such that $c \log(n) \leq c_1 + c_2 \log(n) + c_3 \log(n) + c_4 \log(n) + c_5 \log(n) + c_6$. Dividing by $\log(n)$ and taking the limit as n approaches infinity, we know that c must be less than or equal to $c_2 + c_3 + c_4 + c_5$. Similarly, for the upper bound, there must be a real constant C such that $c_1 + c_2 \log(n) + c_3 \log(n) + c_4 \log(n) + c_5 \log(n) + c_6 \leq C \log(n)$. Doing the same thing, C must be greater than or equal to $c_2 + c_3 + c_4 + c_5$. Since there exist constants c and C and a positive integer n_0 such that $c \log(n) \leq T(n) \leq C \log(n)$ for all $n \geq n_0$, we have shown that $T(n) \in \Theta(\log(n))$.