Cmpe 300: Homework 1 — Due: October 25th 16:00

Solve the following questions in LaTeX or using a word processor. Keep your answers to the main 3 questions in separate pages, though each may span across multiple pages. Deliver a hard copy of your homework to the assistant's mailbox (in the secretary's office) or to his desk in BM 31.

The purpose of this homework is to familiarize you with the complexity related questions. This is an individual homework, so work on your own. Please do not submit just an answer, but show all your reasoning, and how you arrive at the answers. For any further questions, contact the assistant at utkan.gezer@boun.edu.tr.

1. **(30 pts)** Choose the *most precise (smallest)* complexity class among O, Ω, Θ, and $\sim$ making the following statements true. Prove your answer.

   (a) $5^n \in O(7^n)$

   Let a = 5 and b = 7. If n goes to $\infty, the \lim_{n\to\infty} \frac{a^n}{b^n} = \lim_{n\to\infty}((\frac{a}{b})^n)$.
   Therefore, the fraction $(\frac{a}{b})$ is less than one,
   thus $\lim_{n\to\infty}(\frac{a^n}{b^n})$ becomes 0 when $n \to \infty$.

   (b) $5^{n+2} \in \Theta(5^n)$

   5^{n+2} is $5^n \cdot 5^2$ and it is $25 \cdot 5^n$. Since the limit of $\lim_{n\to\infty} \frac{5^n}{5^{n+2}} = \lim_{n\to\infty} \frac{1 \cdot 5^n}{25 \cdot 5^n}$ goes to $\frac{1}{25}$. Hence, it is tightly bounded.

   (c) $\log(n) \cdot \log(n) \in \Omega(\log\log(n))$

   Let $\lim_{n\to\infty} \log(n)$ be y. $\lim_{n\to\infty} \frac{\log(\log(n))}{\log(n)\cdot\log(n)}$ becomes $\lim_{n\to\infty} \frac{\log(y)}{y \cdot y}$and after L'Hospital Rule, the formula becomes $\lim_{n\to\infty} \frac{\frac{y'}{y}}{2 \cdot y \cdot y'}$. And by removing y' from nominator and denominator the formula become $\lim_{n\to\infty} \frac{1}{2 \cdot y^2}$, replacing y with $\lim_{n\to\infty} \log(n)$ simplifies the formula to $\lim_{n\to\infty} \frac{1}{2\cdot(\log(n))^{2]}}$. When $n \to \infty$, the formula $\to \infty$. Hence, it is in the $\Omega\, class of$ $(\log(\log(n)))$

   (d) $\log(n) \in \Omega(\log\log(n))$

   Let $\lim_{n\to\infty} \log(n)$ be y. Replacing it in former equation and take the limit of it becomes into $\lim_{n\to\infty} \frac{\log(y)}{y}$. Applying the L'Hospital Rule gives $\lim_{n\to\infty} \frac{\frac{y'}{y}}{y'}$. Eliminating y' and replacing y with $\lim_{n\to\infty} \log(n)$ makes the final formula as $\lim_{n\to\infty} \frac{1}{\log(n)}$, and the limit goes to 0. Hence, it bounds from lower.

   (e) $\sum_{i=1}^{n} \sqrt{i} \in \Theta\left(\frac{n\sqrt{n}}{2}\right)$

   From $1 \to n$, $\sqrt{i}$ is always less than or equal to $\sqrt{n}$, thus summing it all becomes less than or equal to $n\sqrt{n}$. When n $\to \infty$, the limit $\lim_{n\to\infty} \frac{\frac{n\sqrt{n}}{2}}{n\sqrt{n}}$ goes to $\frac{1}{2}$, which is a constant. Therefore, this summation is tightly bounded with $n\sqrt{n}$.

2. **(35 pts)** Function to the right checks whether n is a prime number in a naive manner.

Assume that the input n also has the size n. So, for example, if $n = 13$, take size of input as 13.

Note *Usually, the size of* n *is taken as* $\log(n)$*, the number of digits, but you should take it as* n *for this question.*

Assume that the **mod** operation takes constant time. Then, give the *most precise* O-class for the worst-case time complexity of this algorithm for when;

Algorithm 1 Primality check (naive)

```
function IsPrime(n)
    for i ← 2 to (n − 1) do
        if (n mod i) = 0 then
            return(.false.)
        end if
    end for
    return(.true.)
end function
```

(a) n is even,

If n is even, n can be fragmented as $2 \cdot l$, where $l > 0$. Thus, at first iteration, it returns true and its worst case complexity is $O(1)$

(b) n is prime,

If n is prime, n cannot be fragmented as a multiplication of two or more numbers. Thus, at each iteration until n, $if(n \mod i) = 0$ returns false. Then, it exits the for loop and returns true. Hence, the worst case complexity of this input is $O(n)$.

(c) n is composite (not prime),

If n is composed of the multiplication of two numbers, in the worst case situation, the numbers should be $\sqrt{n}$. From $2 \rightarrow \sqrt{n}$, the for loop checks for every iteration. In the worst case, the loop executes as $\sqrt{n}$ times, hence the worst case complexity is $O\sqrt{n}$.

(d) and when $n = p \cdot q$, where p and q are primes.

If n is multiplication of two prime numbers, the geometric mean of these two numbers is the worst input that runs the algorithm longer. From $2 \rightarrow \sqrt{n}$, the loop executes and stops at $\sqrt{n}$. Thus the worst case complexity is $O\sqrt{n}$

(e) Give the *most precise* O-class in general (call this $W(n)$).

In the worst case scenario, the algorithm should traverse from $2 \rightarrow (n-1)$, thus $(n-3)$ basic operation will be made. Hence, the $W(n) \in O(n)$

(f) Plot the function $W(n)$ against n.

You are allowed to draw this by hand.
Make sure that your writing is clear!

(g) Plot the function $f(n) = \sqrt{n}$. What is the asymptotic relation between $W(n)$ and $f(n)$?

You are allowed to draw this by hand.
Make sure that your writing is clear!

3. **(35 pts)** Draw the sets $O(n^2)$, $o(n^2)$, $\Omega(n^2)$, $\omega(n^2)$, $\Theta(n^2)$, and $\sim(n^2)$ in a single Venn diagram. Make the boundaries clear for each class. If it the boundary of a class remains uncertain, state which areas in the Venn diagram belongs to that class verbally for clarification.

   Find and write one example member into each region of your diagram. Mark the empty regions with the $\varnothing$ symbol.

You are allowed to draw this by hand.
Make sure that your writing is clear!