Assignment 1

Asymptotic Bounds

Name: Dipam Sen
Roll No: 24CS10059

1.4.1 Prove the transitive property 3 of Big-O:
1. $f = O(g)$ and $g = O(h) \Rightarrow f = O(h)$
2. $f = \Omega(g)$ and $g = \Omega(h) \Rightarrow f = \Omega(h)$

Sol. **1.** We have

$$f = O(g) \Rightarrow \exists\ c_1 > 0, n_1 > 0, f(n) \leq c_1 \cdot g(n) \quad \forall\ n \geq n_1 \tag{1.1}$$

$$g = O(h) \Rightarrow \exists\ c_2 > 0, n_2 > 0, g(n) \leq c_2 \cdot h(n) \quad \forall\ n \geq n_2 \tag{1.2}$$

Then, from (1.1) and (1.2), we have

$$\begin{aligned} &f(n) \leq c_1 c_2 \cdot h(n) \quad \forall\ n \geq \max(n_1, n_2) \\ &\Rightarrow f(n) \leq c \cdot h(n) \quad \forall\ n \geq n_0 \end{aligned}$$

for $c = c_1 c_2$ and $n_0 = \max(n_1, n_2)$.

This shows that $f(n) = O(h(n))$.

2. Similarly, we have

$$f = \Omega(g) \Rightarrow \exists\ c_1 > 0, n_1 > 0, f(n) \geq c_1 \cdot g(n) \quad \forall\ n \geq n_1 \tag{1.3}$$

$$g = \Omega(h) \Rightarrow \exists\ c_2 > 0, n_2 > 0, g(n) \geq c_2 \cdot h(n) \quad \forall\ n \geq n_2 \tag{1.4}$$

Then, from (1.3) and (1.4), we have

$$\begin{aligned} &f(n) \geq c_1 c_2 \cdot h(n) \quad \forall\ n \geq \max(n_1, n_2) \\ &\Rightarrow f(n) \geq c \cdot h(n) \quad \forall\ n \geq n_0 \end{aligned}$$

for $c = c_1 c_2$ and $n_0 = \max(n_1, n_2)$.

This shows that $f(n) = \Omega(h(n))$.

1.4.2

$$\begin{bmatrix} F(n) \\ F(n-1) \end{bmatrix} = M^{n-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ where } M = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

Prove by induction, that $M^n = \begin{bmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{bmatrix}$ $\forall\ n \geq 1$.

Sol. **Base Case:** For $n = 1$, we have

$$M^1 = M = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} F_2 & F_1 \\ F_1 & F_0 \end{bmatrix}$$

which satisfies the proposition.

Inductive Case: We show that if the proposition is true for some $n = k$, it is also true for $n = k + 1$.

Assume that the proposition is true for some $n = k$. Then we have,

$$M^k = \begin{bmatrix} F_{k+1} & F_k \\ F_k & F_{k-1} \end{bmatrix}$$

Now, we can find M^{k+1}:

$$\begin{aligned} M^{k+1} = M \cdot M^k = M\begin{bmatrix} F_{k+1} & F_k \\ F_k & F_{k-1} \end{bmatrix} &= \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} F_{k+1} & F_k \\ F_k & F_{k-1} \end{bmatrix} \\ &= \begin{bmatrix} F_{k+1} + F_k & F_k + F_{k-1} \\ F_{k+1} & F_{k-1} \end{bmatrix} \\ &= \begin{bmatrix} F_{k+2} & F_{k+1} \\ F_{k+1} & F_{k-1} \end{bmatrix} \end{aligned}$$

Thus, the proposition is true for $n = k + 1$.

By the principal of mathematical induction, this shows that the proposition is true for all $n \geq 1$.

1.4.3 A function $f(x)$ is said to be **monotonically increasing** if $f(x) \leq f(y) \forall\ x \leq y$. Suppose $f(n)$ and $g(n)$ are monotonically increasing functions such that in addition, they are non negative. Prove that the following functions are also monotonically increasing:

1. $f(n) + g(n)$
2. $f(g(n))$
3. $f(n) \cdot g(n)$ if $f(n), g(n) > 0$ for all n.

Sol. The definition as given above states that a function f is monotonically increasing if $f(x) \leq f(y) \forall\ x \leq y$. Thus to prove that a function is monotonically increasing, it suffices to show $x \leq y \Rightarrow f(x) \leq f(y)$.

1. Let $x \leq y$. Then,

$$f(x) \leq f(y) \tag{1.5}$$

$$g(x) \leq g(y) \tag{1.6}$$

Adding (1.5) and (1.6) we have

$$f(x) + g(x) \leq f(y) + g(y)$$

thus, $f(n) + g(n)$ is monotonically increasing.

2. Let $x \leq y$. Then, $g(x) \leq g(y)$.

Let $g(x) = k_1, g(y) = k_2$. Clearly, $k_1 \leq k_2$. Since f is monotonically increasing, we have

$$f(k_1) \leq f(k_2)$$
$$\Rightarrow f(g(x)) \leq f(g(y))$$

This, $f(g(n))$ is monotonically increasing.

3. Let $x \le y$. We know that

$$0 < f(x) \le f(y) \tag{1.7}$$

and

$$0 < g(x) \le g(y) \tag{1.8}$$

.

From (1.7), we can write

$$f(x)g(x) \le f(y)g(x)$$

and from (1.8) we can write

$$f(y)g(x) \le f(y)g(y)$$

which together imply

$$f(x)g(x) \le f(y)g(y)$$

thus showing that $f(n)g(n)$ is monotonically increasing.

1.4.4 What is the smallest value of n such that an algorithm whose running time is $100n^2$ runs faster than an algorithm whose running time is 2^n on the same machine?

Sol. Assuming that each operation takes equal time, we can find the threshold value of n where $100n^2 < 2^n$ by trial and error:

$$\begin{aligned}
n = 5 : 100n^2 &= 2500 \quad > 2^n = 32 \\
n = 7 : 100n^2 &= 4900 \quad > 2^n = 128 \\
n = 10 : 100n^2 &= 10000 > 2^n = 1024 \\
n = 12 : 100n^2 &= 14400 > 2^n = 4096 \\
n = 13 : 100n^2 &= 16900 > 2^n = 8192 \\
n = 14 : 100n^2 &= 19600 > 2^n = 16384 \\
\boldsymbol{n = 15} : 100n^2 &= 22500 < 2^n = 32768
\end{aligned}$$

Thus, the algorithm with a running time of $100n^2$ will run faster than the algorithm with a running time of 2^n on the same machine for input sizes $n \ge 15$.

1.4.5 Prove the following:

1. $n! = o(n^n)$

2. $\lg(n!) = \Theta(n \lg n)$ [Prove it in two different ways, one using calculus.]

Sol. **1.** Let $c > 0$ be arbitrary. We need to show that $\exists\ n_0 > 0$ such that

$$n! < c \cdot n^n \quad \forall\ n \ge n_0$$

or equivalently, $c > \dfrac{n!}{n^n}$.

$$\begin{aligned}\Rightarrow \frac{n!}{n^n} &= \frac{1}{n}\cdot\frac{2}{n}\cdot\ldots\cdot\frac{n-1}{n}\cdot\frac{n}{n}\\ &= \prod_{i=1}^{\lfloor n/2\rfloor}\frac{i}{n}\times\prod_{i=\lfloor n/2\rfloor+1}^{n}\frac{i}{n}\end{aligned}$$

Here, for $i = 1$ to $\left\lfloor \frac{n}{2} \right\rfloor$, we have $\frac{i}{n} \le \frac{1}{2}$, and for $i = \left\lfloor \frac{n}{2} \right\rfloor + 1$ to n, $\frac{i}{n} \le 1$.

Thus,

$$\frac{n!}{n^n} \le \left(\frac{1}{2}\right)^{\lfloor n/2\rfloor} \le \left(\frac{1}{2}\right)^{n/2} < c$$

$$\begin{aligned}\left(\frac{1}{2}\right)^{n/2} < c \Rightarrow -\frac{n}{2} &< \lg c\\ \Rightarrow n &> -2\lg c\end{aligned}$$

Thus, for any arbitrary $c > 0$, we can choose $n_0 = \max(-2\lg c, 0)$, such that $n! < c \cdot n^n \ \forall\ n > n_0$. This shows that $n! = o(n^n)$.

Method 2:

We can directly use the Stirling approximation:

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n\left(1 + \Theta\left(\frac{1}{n}\right)\right)$$

Consider $f(n) = n!$, $g(n) = n^n$.

Then

$$\lim_{n\to\infty}\frac{f(n)}{g(n)} = \lim_{n\to\infty}\frac{n!}{n^n} = \lim_{n\to\infty}\frac{\sqrt{2\pi n}}{e^n}\left(1 + \Theta\left(\frac{1}{n}\right)\right) = 0$$

Thus, $n! = o(n^n)$.

2. We have,

$$\lg(n!) = \sum_{i=1}^{n}\lg(i) \le \sum_{i=1}^{n}\lg(n) \le n\lg n$$

This shows that $\lg(n!) = O(n\lg n)$.

Also,

$$\begin{aligned}\lg(n!) &= \sum_{i=1}^{n}\lg(i)\\ &= \sum_{i=1}^{\lceil n/2\rceil}\lg(i) + \sum_{i=\lceil n/2\rceil+1}^{n}\lg(i)\end{aligned}$$

Here, for $i = 1$ to $\left\lceil \frac{n}{2} \right\rceil$, $\lg(i) \ge 0$, and for $i = \left\lceil \frac{n}{2} \right\rceil + 1$ to n, $\lg(i) \ge \lg\left(\frac{n}{2}\right)$.

Thus,

$$\begin{aligned}
\lg(n!) &\geq \sum_{i=\lceil n/2\rceil+1}^{n} \lg\left(\frac{n}{2}\right) = \left\lfloor\frac{n}{2}\right\rfloor \lg\left(\frac{n}{2}\right) \geq \left(\frac{n}{2}-1\right)\lg\left(\frac{n}{2}\right) \\
&\geq \left(\frac{1}{2}n-1\right)(\lg n - 1) = \frac{1}{2}n\lg(n) - \frac{n}{2} - \lg(n) + 1 \\
&\geq \frac{1}{2}n\lg(n) + 1 \geq \Theta(n\lg n)
\end{aligned}$$

This shows that $\lg(n!) = \Omega(n\lg n)$.

Together, this implies that $\lg(n!) = \Theta(n\lg n)$.

Method 2:

We can apply Stirling's approximation:

$$\begin{aligned}
n! &= \sqrt{2\pi n}\left(\frac{n}{e}\right)^n\left(1+\Theta\left(\frac{1}{n}\right)\right) \\
\Rightarrow \lg(n!) &= \lg\sqrt{2\pi} + \frac{1}{2}\lg n + n\lg n - n\lg e + \lg\left(1+\Theta\left(\frac{1}{n}\right)\right) \\
&= \Theta(n\lg n)
\end{aligned}$$

Method 3:

Consider $\lg(n!) = \sum_{i=2}^{n}\lg(i)$. Since $\lg(x)$ is monotonically increasing, this sum is upper bounded by the integral $\int_2^{n+1}\lg(x)\,\mathrm{d}x$, and lower bounded by the integral $\int_1^{n}\lg(x)\,\mathrm{d}x$. This is clear from the below figures.

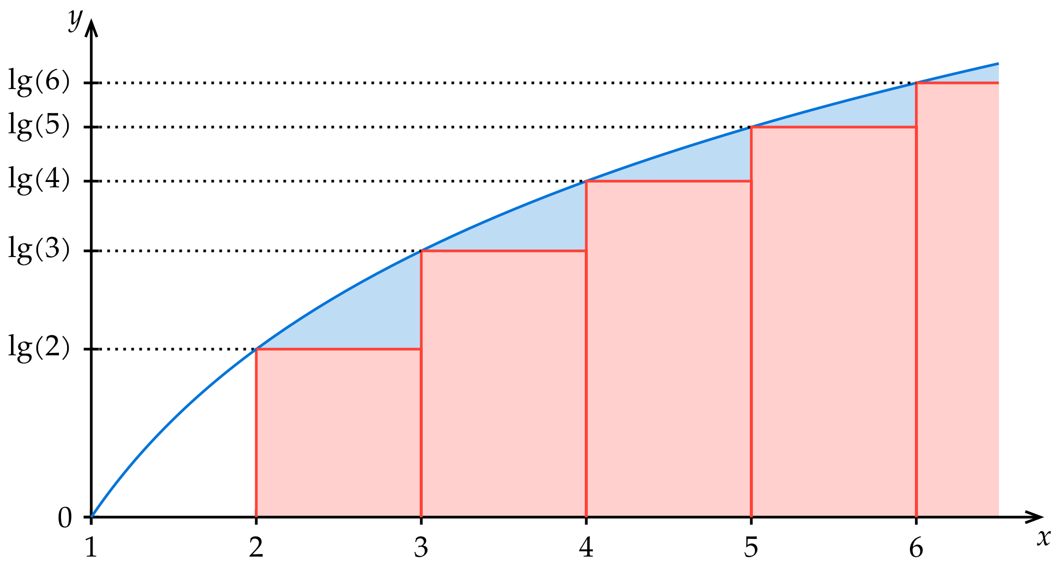

Figure 1: Plot showing $\sum_{i=2}^{n}\lg(i) < \int_2^{n+1}\lg(x)\,\mathrm{d}x$. The blue plot represents $\lg(x)$ and the red rectangles represent the discrete sum.

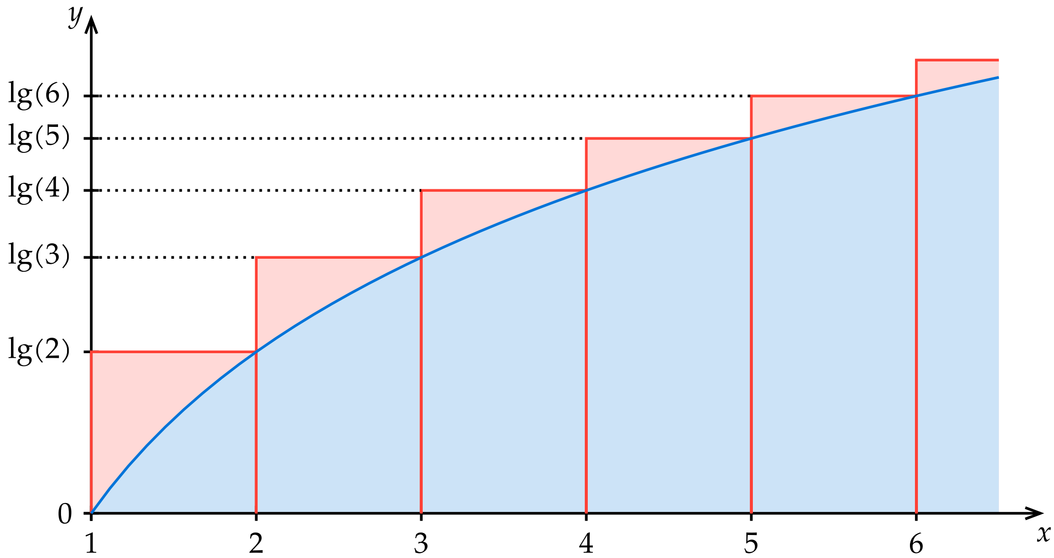

Figure 2: Plot showing $\sum_{i=2}^{n} \lg(i) > \int_1^n \lg(x)\,\mathrm{d}x$. The blue plot represents $\lg(x)$ and the red rectangles represent the discrete sum. Note that here the red rectangles are shifted left, as compared to the previous figure.

We can give a general statement for the above observation:

For a continuous monotonically increasing function $f(x)$, we have

$$\int_{a-1}^{b} f(x)\,\mathrm{d}x \le \sum_{i=a}^{b} f(i) \le \int_{a}^{b+1} f(x)\,\mathrm{d}x$$

Also, the reverse inequality holds for a monotonically decreasing function.

Thus,

$$\int_1^n \lg(x)\,\mathrm{d}x \le \sum_{i=2}^{n} \lg(i) \le \int_2^{n+1} \lg(x)\,\mathrm{d}x$$

We can simplify the integrals:

$$\int \lg(x)\,\mathrm{d}x = \frac{1}{\log(2)} \int \log(x)\,\mathrm{d}x = \frac{1}{\log(2)}(x \log x - x)$$

So we have,

$$\begin{aligned} \frac{1}{\log(2)}(n \log n - n + 1) &\le \lg(n!) \\ &\le \frac{1}{\log(2)}((n+1)\log(n+1) - n - 2\log 2 + 2) \\ \Rightarrow \Theta(n \lg n) \le \lg(n!) &\le \Theta(n \lg n) \end{aligned}$$

Hence, this shows that $\lg(n!) = \Theta(n \lg n)$.