Miray YILDIZ
161044023

CSE 321 - Introduction to Algorithms - Fall 2020
Homework 1

1) a) $\underbrace{\log_2 n^2+1}_{f(n)} \in O(\underbrace{n}_{g(n)})$

To be $\in O(n)$, it will be $f(n) \leq c \cdot g(n)$ whenever $n \geq n_0$

$\log_2 n^2+1 \leq c.n$

Let $\begin{cases} c=2 \\ n_0=1 \end{cases}$ $\quad \log_2 n^2+1 \leq 2n$ for $n \geq 1$ ✓ True for all $n \geq 1$

So, $\log_2 n^2+1 \in O(n)$ is true.

b) $\underbrace{\sqrt{n(n+1)}}_{f(n)} \in \Omega(\underbrace{n}_{g(n)})$

To be $\in \Omega(n)$, it will be $f(n) \geq c.g(n)$ whenever $n \geq n_0$

$\sqrt{n(n+1)} \geq c.n$

Let $\begin{cases} c=1 \\ n=1 \end{cases}$ $\quad \sqrt{n(n+1)} \geq n$ for $n \geq 1$ ✓ True for all $n \geq 1$

So, $\sqrt{n(n+1)} \in \Omega(n)$ is true.

c) $\underbrace{n^{n-1}}_{f(n)} \in \Theta(\underbrace{n^n}_{g(n)})$

To be $\in \Theta(n^n)$, it will be $c_1.g(n) \leq f(n) \leq c_2.g(n)$ whenever $n \geq n_0$

$c_1.(n^n) \leq n^{n-1} \leq c_2.(n^n)$

Let $\begin{cases} c_1=1/3 \\ c_2=1 \\ n_0=3 \end{cases}$ $\quad \frac{1}{3}.n^n \leq n^{n-1} \leq 1.n^n$ $\quad n=3$ true but $n=4$ is false.

So, $n^{n-1} \in \Theta(n^n)$ is not true, it is false.

d) $O(2^n+n^3) \subset O(4^n)$

If $O(2^n+n^3) \subset O(4^n)$, $(2^n+n^3) \in O(4^n)$

$f(n) \leq c.g(n)$, $n \geq n_0$

$2^n+n^3 \leq c.4^n$

Let $\begin{cases} c=1 \\ n_0=0 \end{cases}$ $2^n+n^3 \leq 1.4^n$ for $n \geq 0$ ✓ true for all $n \geq 0$

So, $O(2^n+n^3) \subset O(4^n)$ when $n \geq 1$

e) $O(2\log_3 \sqrt[3]{n}) \subset O(3\log_2 n^2)$

If $O(2\log_3 \sqrt[3]{n}) \subset O(3\log_2 n^2)$, $\underbrace{(2\log_3 \sqrt[3]{n})}_{f(n)} \in O(\underbrace{3\log_2 n^2}_{g(n)})$

$f(n) \leq c.g(n)$, $n \geq n_0$

$2\log_3 \sqrt[3]{n} = 3\log_3 n$

$3\log_2 n^2 = 6\log_2 n$

$3\log_3 n \leq c.6\log_2 n$

Let $\begin{cases} n_0=1 \\ c=1 \end{cases}$ $3\log_3 n \leq 1.6\log_2 n$, for all $n \geq 1$ ✓ true for all $n \geq 1$

So, $O(2\log_3 \sqrt[3]{n}) \subset O(3\log_2 n^2)$

f) $\log_2 \sqrt{n}$ and $(\log_2 n)^2$ are of the same asymptotical order?

$$\lim_{n\to\infty} \frac{\log_2\sqrt{n}}{(\log_2 n)^2} = \lim_{n\to\infty} \frac{(\log_2\sqrt{n})'}{(\log_2 n)^{2'}} \rightarrow \text{(L' hopital Rule)}$$

$$\lim_{n\to\infty} \frac{\frac{1}{2\cancel{x\ln 2}}}{\frac{\log_2(x^2)}{\cancel{x\ln 2}}} = \lim_{n\to\infty} \frac{\frac{1}{2}}{\log_2(x^2)} \Rightarrow \lim_{n\to\infty} \frac{1}{2\log_2(x^2)} = 0$$

Due to result is 0, $\log_2\sqrt{n}$ is smaller than $(\log_2 n)^2$, they are not same asymptotical order. It is false.

2) $n^2, n^3, n^2 \log n, \sqrt{n}, \log_2 n, 10^n, 2^n, 8^{\log_2 n}$

$\propto$ First, exponential functions grow faster than others. So, 10^n and 2^n are faster than others. $10>2$, so 10^n is faster than 2^n.

$\propto$ logs grow slower than roots. So, $\log_2 n$ grow slower than $\sqrt{n}$.

$\propto$ Roots grow slower than polynomials. So, n^3, $n^2 \log_2 n$ and n^2 grow faster than $\sqrt{n}$.

According to this informations, result is

$$10^n > 2^n > n^3 = 8^{\log_2 n} > n^2 \log_2 n > n^2 > \sqrt{n} > \log_2 n$$

3)

a) If my_array has 1 element, complexity of program is $\theta(1)$, because for loop run 1 time. But my_array has n element and $n>1$, complexity of program is $\theta(n)$, because loop run n times.

$\left.\begin{array}{l}\text{Worst case is } \theta(n) \\ \text{Best case is } \theta(1)\end{array}\right\}$ Complexity of program is $\underline{\underline{O(n)}}$

b)

4)

a) $\sum_{i=1}^{n} i^2 \log i$ it is non-decreasing.

Since it is non decreasing,

$$\int_0^n g(x)\,dx \leq f(n) \leq \int_1^{n+1} g(x)\,dx$$

$$\int_0^n x^2 \log x\,dx \leq f(n) \leq \int_1^{n+1} x^2 \log x\,dx$$

Solution of integral:

$u = \log x \qquad v = x^2$

$$\Rightarrow \log x \int x^2 dx - \int \left(\frac{d(\log x)}{dx} \int x^2 dx \right) dx \Rightarrow \frac{x^3}{3} \log x - \frac{1}{3} \int \frac{1}{x} x\,x^3 dx$$

$$\Rightarrow \frac{x^3}{3} \log x - \frac{1}{3} \int x^2 dx \rightarrow \frac{x^3}{3} \log x - \frac{x^3}{9} = \frac{x^3}{3} \left(\log x - \frac{1}{3} \right)$$

$$\frac{x^3}{3} \left(\log x - \frac{1}{3} \right) \Big|_0^n \leq f(n) \leq \frac{x^3}{3} \left(\log x - \frac{1}{3} \right) \Big|_1^{n+1}$$

$$= \frac{n^3}{3} \left(\log n - \frac{1}{3} \right) \leq f(n) \leq \frac{(n+1)^3 \left(\log(n+1) - \frac{1}{3} \right) - 1}{3}$$

$$\left. \begin{array}{l} f(n) \in O(n^3 \log n) \\ f(n) \in \Omega(n^3 \log n) \end{array} \right\} f(n) \in \Theta(n^3 \log n)$$

b) $\sum_{i=1}^{n} i^3$ nondecreasing

$$\int_0^n g(x)\,dx \leq f(n) \leq \int_1^{n+1} g(x)\,dx$$

$$\int_0^n x^3 dx \leq f(x) \leq \int_1^{n+1} x^3 dx$$

$$= \frac{x^4}{4}\Big|_0^n \leq f(n) \leq \frac{x^4}{4}\Big|_1^{n+1} \Rightarrow \frac{n^4}{4} \leq f(n) \leq \frac{(n+1)^4 - 1}{4}$$

$$\left.\begin{aligned} f(n) &\in O(n^4) \\ f(n) &\in \Omega(n^4) \end{aligned}\right\} \underline{\underline{f(n) \in \Theta(n^4)}} \checkmark$$

c) $\sum_{i=1}^{n} 1/(2\sqrt{i})$ it is non increasing

$$\int_1^{n+1} g(x)\,dx \leq f(n) \leq \int_0^n g(x)\,dx$$

$$\int_1^{n+1} 1/(2\sqrt{i})dx \leq f(n) \leq \int_0^n 1/(2\sqrt{i})dx$$

(Solution of integration:

$$\frac{1}{2}\int \frac{1}{\sqrt{x}}dx \rightarrow \frac{1}{2}\int x^{-\frac{1}{2}}dx \rightarrow \frac{1}{2}\left(\frac{x^{\frac{1}{2}}}{\frac{1}{2}}\right) = \underline{\underline{x^{\frac{1}{2}}}} = \underline{\underline{\sqrt{x}}}$$)

$$\sqrt{x}\Big|_1^{n+1} \leq f(n) \leq \sqrt{x}\Big|_0^n \Rightarrow \sqrt{n+1} - 1 \leq f(n) \leq \sqrt{n}$$

$$\left.\begin{aligned} f(n) &\in O(\sqrt{n}) \\ f(n) &\in \Omega(\sqrt{n}) \end{aligned}\right\} f(n) \in \Theta(\sqrt{n}) \checkmark$$

d) $\sum_{i=1}^{n} 1/i$ it is non increasing

$$\int_1^{n+1} g(x)\,dx \leq f(n) \leq \int_0^n g(x)\,dx$$

$$\int_1^{n+1} 1/i\,dx \leq f(n) \leq \int_0^n 1/i\,dx$$

(Solution of integration:
$\int 1/x = \ln x + C$)

$$\ln x \Big|_1^{n+1} \leq f(n) \leq \ln x \Big|_0^n$$

$$\ln(x+1) \leq f(n) \leq \underbrace{\ln(n) - \ln 0}_{\infty}$$

$$f(n) = \Omega(\ln n) \checkmark$$

5)

The worst case is when the searched Item is not in the array. In this case, the entire array is searched. The search process is repeated n(array size). So, the worst case is $\Theta(n)$.

In the best case analysis, we examine the situation where there is minimum action. The best case in linear search is when the searched element is in the first element of the array. The program ends after the element is found. Based on this, the base case would be $\Theta(1)$.