CSE 321 - Introduction to Algorithm Design

Serval MEHDER
151044009

- Homework 1 -

Q1

$T_1(n) = 3n^4 + 3n^3 + 1$

$T_2(n) = 3^n$

$T_3(n) = (n-2)!$

$T_4(n) = \ln^2 n$

$T_5(n) = 2^{2n}$

$T_6(n) = \sqrt[3]{n}$

→ Order them according to their asymptotic complexity.

$T_4 < T_6 < T_1 < T_2 < T_5 < T_3$

→ Then prove with using limits that, every function has more complexity from the function to its left one.

- $T_4 - T_6$

$$\lim_{n\to\infty} \frac{\ln^2 n}{\sqrt[3]{n}} \;\; \frac{\infty}{\infty} \quad \text{L'Hospital} \rightarrow \lim_{n\to\infty} \frac{\frac{2\ln n}{n}}{\frac{1}{3\cdot\sqrt[3]{n^2}}} = \lim_{n\to\infty} \frac{6\ln n}{\sqrt[3]{n}} \;\; \frac{\infty}{\infty}$$

$$\text{L'Hospital} \rightarrow \lim_{n\to\infty} \frac{\frac{6}{n}}{\frac{1}{3\cdot\sqrt[3]{n^2}}} = \lim_{n\to\infty} \frac{18}{\sqrt[3]{n}} = \underline{\underline{0}}$$

$T_4 = O(T_6)$ ✓

- $T_6 - T_1$

$$\lim_{n\to\infty} \frac{\sqrt[3]{n}}{3n^4+3n^3+1} = \lim_{n\to\infty} \frac{\sqrt[3]{n} \,/\, n^{\frac{1}{3}}}{(3n^4+3n^3+1)\,/\,n^{\frac{1}{3}}} = \frac{1}{3n^{\frac{4}{3}} + \cdots} = \frac{\text{sayı}}{\infty} = \underline{\underline{0}}$$

$T_6 = O(T_1)$ ✓

- $T_1 - T_2$

$$\lim_{n\to\infty} \frac{3n^4+3n^3+1}{3^n} \;\; \frac{\infty}{\infty} \quad \text{L'Hospital} \rightarrow \frac{12n^3+9n^2}{3^n \ln 3} \;\; \frac{\infty}{\infty} \quad \text{L'Hospital} \rightarrow \frac{36n^2+18n}{\ln^2 3 \cdot 3^n} \;\; \frac{\infty}{\infty}$$

$$\text{L'Hospital} \rightarrow \frac{72n+18}{3^n \cdot \ln^3 3} \;\; \frac{\infty}{\infty} \quad \text{L'Hospital} \rightarrow \frac{72}{3^n \cdot \ln^4 3} = \frac{\text{sayı}}{\infty} = \underline{\underline{0}}$$

$T_1 = O(T_2)$ ✓

- $T_2 - T_5$

$$\lim_{n\to\infty} \frac{3^n}{2^{2n}} = \lim_{n\to\infty} \frac{3^n}{(2^2)^n} = \lim_{n\to\infty} \left(\frac{3}{4}\right)^n$$ $0<\frac{3}{4}<1$ olduğu için giderek azalır ve sıfıra yaklaşır $= 0$

$T_2 = O(T_5)$ ✓

- $T_5 - T_3$

$$\lim_{n\to\infty} \frac{2^{2n}}{(n-2)!} = \lim_{n\to\infty} \frac{2^{2n} = (2^2)^n}{\sqrt{2\pi n}\cdot\left(\frac{n}{e}\right)^n}$$ Paydada n^n var 4^n den daha hızlı büyür $= 0$

$T_5 = O(T_3)$ ✓

Q_2 → Explain what this algorithm does. Tell the role of all variables.

watermelon = max value
plum = min value
orange = max ve min'in ortalamasına en yakın değer
orange Time = max ve min değerleri bulma işleminin bitip bitmediğini söyleyen kontrol ifadesi
fruit = Döngünün her adımında sıradaki indexteki değeri gösteren değişken
fruits = input array

Algoritma, array'deki maksimum ve minimum değerleri bulduktan sonra bu iki değerin ortalamasına en yakın ve array'deki en sağdaki (indexi en büyük) değeri return eder.

worst case: − (n-1) kere sağ shift yapılmış,
− 1 kere sol shift yapılmış
yani en küçük eleman en sondaysa worst case olur.

$W(n) = n \in \Theta(n)$

best case: Hiç shift yapılmadan küçükten büyüğe bir liste varsa best case meydana gelir.

$B(n) = n \in \Theta(n)$

Avarage case: Best case ile worst case eşit olduğu için average case'de aynıdır.

$A(n) \in \Theta(n)$

Q3 a. $\sum_{i=0}^{n-1}(i^2+1)^2 = \sum_{i=0}^{n-1} i^4+2i^2+1$

$$\int_0^n g(n) \le f(n) \le \int_1^{n+1} g(n)$$

$$\int_0^n (i^4+2i^2+1)\,di \le f(n) \le \int_1^{n+1} (i^4+2i^2+1)\,di$$

$$\frac{i^5}{5}+\frac{2i^3}{3}+i\Big|_0^n \le f(i) \le \frac{i^5}{5}+\frac{2i^3}{3}+i\Big|_1^{n+1}$$

$$\ldots n^5 \ldots \le f(n) \le \ldots n^5 \ldots$$ sonuç olarak $f(n)=\Theta(n^5)$

b. $\sum_{i=2}^{n-1}\log i^2 = 2.\sum_{i=2}^{n-1}\log i = \underbrace{\log 2+\log 3+\ldots+\log(n-1)}_{n-3 \text{ eleman}}$

Her eleman $\log(n-1)$ den küçük eşit

$$\le (n-3).\log(n-1)$$

$$\in O(n\log n) \rightarrow \text{upper}$$

$$L(n) = 2\sum_{i=2}^{\lceil\frac{n-1}{2}\rceil}\log i + 2.\sum_{i=\lceil\frac{n-1}{2}+1\rceil}^{n-1}\log i \ \ge \sum_{i=\lceil\frac{n-1}{2}+1\rceil}^{n-1}\log i$$

$\lceil\frac{n-3}{2}\rceil$ eleman var, hepsi $\log\left(\frac{n-1}{2}\right)$ den büyük eşit

$$L(n) \ge \frac{n-3}{2}.\log\left(\frac{n-1}{2}\right) \longrightarrow L(n)\in\Omega(n\log n) \rightarrow \text{lower}$$

— sonuç olarak $f(n)=\Theta(n\log n)$

c. $\sum_{i=1}^{n}(i+1)2^{i-1}$

$$\int_0^n (i+1).2^{i-1}.di \le f(n) \le \int_1^{n+1}(i+1).2^{i-1}.di$$

$2^{i-1}.di = dv$ $i+1=u$
$\frac{2^{i-1}}{\ln 2}=v$ $di=du$

$$(i+1).\frac{2^{i-1}}{\ln 2} - \underbrace{\int_0^n \frac{2^{i-1}}{\ln 2}.di}_{\frac{2^{i-1}}{\ln^2 2}}\Big|_0^n \le f(n) \le (i+1).\frac{2^{i-1}}{\ln 2} - \int_0^n \frac{2^{i-1}}{\ln 2}\,di\Big|_1^{n+1}$$

$$\frac{2^{i-1}}{\ln^2 2}\left((i+1).\ln 2-1\right)\Big|_0^n \le f(n) \le \frac{2^{i-1}}{\ln^2 2}\left((i+1).\ln 2-1\right)\Big|_1^{n+1}$$

$$\ldots 2^{n-1}.(n+1)\ldots \le f(n) \le 2^{n-1}.(n+2)\ldots$$

... sonuç olarak $f(n)=\Theta(n.2^n)$

$$d.\ \sum_{i=0}^{n-1}\sum_{j=0}^{i-1}(i+j) = \sum_{i=0}^{n-1}\underbrace{(i+(i+1)+(i+2)+\ldots+(2i-1))}$$

$$\frac{\text{son terim} - \text{ilk terim}}{\text{ortak fark}} + 1 = \frac{2i-1-i}{1} + 1 = i \rightarrow \text{eleman sayısı}$$

ortalama

$$\frac{2i-1+i}{2}\cdot i = \frac{3i-1}{2}\cdot i = \frac{(3i-1)\cdot i}{2}$$

$$\sum_{i=0}^{n-1}\frac{(3i-1)\cdot i}{2} = \frac{1}{2}\cdot\sum_{i=0}^{n-1}3i^2 - i$$

$$\int_0^n (3i^2-i)\,di \le f(n) \le \int_1^{n+1}(3i^2-i)\,di$$

$$i^3 - \frac{i^2}{2}\Big|_0^n \le f(n) \le i^3 - \frac{i^2}{2}\Big|_1^{n+1}$$

$$\_\, n^3 \,\_\_ \le f(n) \le \_\, n^3 \,\_$$

— sonuç olarak $f(n) = \Theta(n^3)$

→ Find the order of growth of each of term.

$n\log n < n^3 < n^5 < 2^n \cdot n$

$b < d < a < c$

→ write corresponding C or Python code at first (a) and the last (d) one.

for a:

```
def third-a(n):
    sum = 0
    for i in range(n):
        sum += ((i**2)+1)**2
    return sum

# for n=5 output is 419
```

for d:

```
def third-d(n):
    sum = 0
    for i in range(n):
        for j in range(i):
            sum += (i+j)
    return sum

# for n=5 output is 40
```

Q4)

ilk adımda n kere çalışır
ikinci " $n/2$ " "
üçüncü " $n/4$ " "
⋮
Son " 1 kere çalışır.

$$= n + \frac{n}{2} + \frac{n}{4} + \dots + 1$$

$$= \sum_{i=0}^{\log_2 n} \frac{n}{2^i} = n \cdot \underline{\sum_{i=0}^{\log_2 n} \frac{1}{2^i}}$$

complexity = $O(n)$

$T(n) = O(n + \frac{n}{2} + \frac{n}{4} + \dots + 1)$

$T(n) \in \underline{O(n)}$

Q5)

a. $n^3 \in O(3^{2n})$

$f(n) \le c \cdot g(n) \quad n \ge n_0$

$n^3 \le c \cdot 3^{2n} \quad n \ge n_0$

$c = 1$ ve $n_0 = 1$ için $\rightarrow n^3 \le 3^{2n}, \ n \ge 1$

Sağlayan c ve n_0 bulduğumuza göre $n^3 \in O(3^{2n})$ ✓

b. $n \in o(\log \log n)$

$$\lim_{n\to\infty} \frac{n}{\lg \lg n} \stackrel{\infty}{=} \text{L'Hospital} \rightarrow \lim_{n\to\infty} \frac{1}{\frac{1}{n \lg n}} = \lim_{n\to\infty} n \cdot \lg n = \infty \cdot \infty = \infty$$

$n \in \omega(\lg \lg n)$
$n \notin o(\lg \lg n)$ ✗

c. $n^2 \lg^2 n \in O(n!)$

$n^2 \lg^2 n \le c \cdot n! \quad n \ge n_0$

$c = 1$ ve $n_0 = 1$ için $\rightarrow n^2 \lg^2 n \le n! \quad n \ge 1$

sağlayan $c = 1$ ve $n_0 = 1$ bulduğumuza göre $n^2 \lg^2 n \in O(n!)$ ✓

d. $\sqrt{10n^2 + 7n + 3} \in \Theta(n)$

$$\lim_{n\to\infty} \frac{\sqrt{10n^2+7n+3}}{n} = \lim_{n\to\infty} \sqrt{\frac{10n^2+7n+3}{n^2}} = \lim_{n\to\infty} \sqrt{10 + \cancel{\frac{7}{n}}^{0} + \cancel{\frac{3}{n^2}}^{0}} = \sqrt{10} \rightarrow \underline{\text{constant}}$$

$\sqrt{10n^2+7n+3} \in \Theta(n)$ ✓