25.10.2024

BLM 2425 ALGORİTMA ANALİZİ

ÖDEV 1

Asymptotic Analysis, Mathematical Analysis of Non-Recursive and Recursive Problems

Öğrenci Adı: Sinem SARAK
Öğrenci Numarası: 22011647
Dersin Eğitmeni: M. Elif KARSLIGİL

1- Master Theorem

Master Teoreminin formülü aşağıdaki gibidir:

Verilmiş olan bir T(n) rekürans bağıntısında

$$T(n) = a.T(\tfrac{n}{b}) + f(n) \quad ve \quad T(1) = c$$

$a \geq 1$,$b > 1$ ve $c > 0$ değerleri için $d \geq 0$ durumunda $f(n) \in \Theta(n^d)$ olmak üzere:

$$T(n) \in \begin{cases} \Theta(n^d) & \text{if } a < b^d \\ \Theta(n^d \log_b n) & \text{if } a = b^d \\ \Theta(n^{\log_b a}) & \text{if } a > b^d \end{cases}$$

şeklinde incelenir. Bu formül kullanılarak verilen sorular alt başlıklarda incelenecektir.

a. $T(n) = 9\, T(n/4) + n^2$

Verilen sorudaki bağıntıda yer alan değerler formüldeki yerlerine konulduğunda a = 9, b = 4 ve d = 2 olduğu görülür. Buna göre b^d ile a arasındaki ilişki incelendiğinde:

$b^d = 4^2 = 16$ ve $a = 9$ olduğuna göre $b^d > a$ dır.

Bu durumda case 1 geçerli olmaktadır.

$$\boxed{T(n) \in \Theta(n^2)}$$

b. T(n) = 3 T(n/3) + log (n)

Verilen sorudaki bağıntıda yer alan değerler formüldeki yerlerine konulduğunda a = 3 ve b = 3 olduğu görülür. f(n) ifadesi log(n) olduğundan d değeri için tam sayı bir ifade kullanılamaz. Ancak d değerinin aralık değerleri kullanılarak master teoremi uygulanabilir:

$n^d = \log(n) < n$ ifadesi yazılabilir. $n^d < n$ olduğuna göre d < 1 denilebilir. Buna göre: $b^d = 3^d < 3$ olduğu söylenebilir. $a = 3$ olduğuna göre $b^d < a$ olur ve case 3'e karşılık gelir.

$$\log_b a = \log_3 3 = 1 \longrightarrow \boxed{T(n) \in \Theta(n)}$$

c. T(n) = 3 T(n/2) + n

Verilen sorudaki bağıntıdaki değerler formüldeki yerlerine konulduğunda a = 3, b = 2 ve d = 1 olduğu görülür. Buna göre b^d ile a arasındaki ilişki incelendiğinde:

$b^d = 2^1 = 2$ ve $a = 3$ ise $2 < 3$ ve $b^d < a$'dır.

Bu durumda case 3 geçerli olmaktadır.

$$\log_b a = \log_2 3 \longrightarrow \boxed{T(n) \in \Theta(n^{\log_2 3})}$$

2- Karmaşıklıkları Big-Oh Cinsinden İfade Etme

1. f1()

```
int f1(int N) {
        int x = 0;
            for (int i = 0; i < N; i++)
                x++;
return x;
}
```

$\}$ n defa döner.

Verilen fonksiyonun basic operation'ı n defa dönen bir döngüdür. Karmaşıklığı $\boxed{f1() \in O(n)}$ olarak ifade edilir.

2. f2()

```
int f2(int N) {
int x = 0;
        for (int i = 0; i < N; i++)
                for (int j = 0; j < i; j++)
                    x += f1(j);
return x;
}
```

→ n adet işlem

→ i adet işlem

→ j adet çağrı

Tüm ifade toplandığında $\sum_{i=0}^{n} \sum_{j=0}^{i} \sum_{k=0}^{j}$ adet çağrı işlemi gerçekleşir. Buna göre Toplam işlem sayısı

$$\sum_{i=0}^{n} \sum_{j=0}^{i} \sum_{k=0}^{j} 1 = \frac{1}{6} . (n+1) . (n+2) . (n+3)$$

olarak hesaplanır.

$\boxed{f2() \in O(n^3)}$ olarak bulunur.

3. f3()

```
int f3(int N) {
        if (N == 0) return 1;
int x = 0;
        for (int i = 0; i < N; i++)
            x += f3(N-1);
return x;
}
```

Recursive bir şekilde çalışan bu fonksiyon için rekürans bağıntısı şu şekilde yazılır:

$$T(n) = \underset{\downarrow}{n} . T(n-1)$$

döngüden kaynaklı, fonksiyon kendini n kere çağırır

$$T(n) = n . T(n-1) = n(n-1) . T(n-2) = \underbrace{n . (n-1) . (n-2)}_{n!} . T(n-3)$$

$$T(n-1) = (n-1) . T(n-2)$$

$$T(n-2) = (n-2) . T(n-3)$$

$\vdots$

ifade n! oranında büyür. $\boxed{f3() \in O(n!)}$

4. f4()

```
int f4(int N) {

if (N == 0) return 0;

return f4(N/2) + f1(N) + f1(N) + f1(N) +
f4(N/2);
}
```

↓ n ↓ n ↓ n

Recursive bir şekilde çalışan bu fonksiyon için rekürans bağıntısı şu şekilde yazılır:

$$T(n) = 2T\left(\frac{n}{2}\right) + 3n = 4T\left(\frac{n}{4}\right) + 6n = 8T\left(\frac{n}{8}\right) + 9n$$

$$T\left(\frac{n}{2}\right) = 2T\left(\frac{n}{4}\right) + 3\frac{n}{2}$$

$$T\left(\frac{n}{4}\right) = 2T\left(\frac{n}{8}\right) + 3\frac{n}{4}$$

$\vdots$

$$T(n) = 2^i T\left(\frac{n}{2^i}\right) + 3in \xrightarrow{n = 2^i \text{ için}} 2^i T\left(\frac{2^i}{2^i}\right) + 3 . 2^i . i$$

$$= \underbrace{n . T(1)}_{} + \boxed{3 . n . \log_2 n}$$

T(1) sabit bir sayı çıkacağından diğer ifade daha kapsamlıdır

$\boxed{f4() \in O(n \log n)}$

3- Tablo

Soru	Cevap	f(n)	g(n)
1	O	n^2	n^3
2	Ω	n lg n	n
3	Θ	1	3 + sin n
4	Ω	3^n	2^n
5	Θ	4^{n+4}	2^{2n+2}
6	O	n lg n	$n^{105/100}$
7	Θ	$\lg \sqrt{10n}$	$\lg n^3$
8	O	n!	(n+1)!

1) $f(n) \in ?(g(n))$

$$g(n).c_2 \leq f(n) \leq g(n).c_1$$

$$n^3.c_2 \leq n^2 \leq n^3.c_1$$

$n_0 > 0$ ve $c_2 > 0$ olmak üzere burası sağlanamaz

$n^2 \leq n^3.c_1$

$n_0 \geq 1$ ve $c_1 = 2$ için sağlanır

$$\boxed{f(n) \in O(g(n))}$$

2) $f(n) \in ?(g(n))$

$$g(n).c_2 \leq f(n) \leq g(n).c_1$$

$$n.c_2 \leq n.\log n \leq n.c_1$$

$c_2 \leq \log n$

$n_0 = 2$ $c_2 = 1$ için sağlanır

$\log n \leq c_1$

$n \to \infty$ durumunda bu eşitlik sağlanamaz

$$\boxed{f(n) \in \Omega(g(n))}$$

3) $f(n) \in ?(g(n))$

$$g(n).c_2 \leq f(n) \leq g(n).c_1$$

$$c_2(3+\sin(n)) \leq 1 \leq c_1(3+\sin(n))$$

$3 + \sin(n) \leq \frac{1}{c_2}$

$\sin(n) \leq \frac{1}{c_2} - 3$

$n_0 = 90$ ve $c_2 = \frac{1}{4}$ için bu kısım sağlanır.

$\frac{1}{c_1} \leq 3 + \sin(n)$

$\frac{1}{c_1} - 3 \leq \sin(n_0)$

$c_1 = 1$ ve $n_0 = 90$ için burası sağlanır

$$\boxed{f(n) \in \Theta(g(n))}$$

4) $f(n) \in ?(g(n))$

$$g(n).c_2 \leq f(n) \leq g(n).c_1$$

$$2^n.c_2 \leq 3^n \leq 2^n.c_1$$

$n_0 > 0$ olduğundan $2^n < 3^n$ c_2'nin herhangi bir değeri için sağlanır.

$2^n < 3^n$ olduğundan sağlanamaz

$$\boxed{f(n) \in \Omega(g(n))}$$

5) $f(n) \in ?(g(n))$

$$g(n).c_2 \leq f(n) \leq g(n).c_1$$

$$2^{2n+2}.c_2 \leq 4^{n+1} \leq 2^{2n+2}.c_1$$

$2^{2(n+1)}.c_2 \leq 4^{n+1}$

$4^{n+1}.c_2 \leq 4^{n+1}$

$c_2 \leq 1$ $c_2 = 1$ için sağlanır

$4^{n+1} \leq 2^{2(n+1)} c_1$

$4^{n+1} \leq 4^{n+1}.c_1$

$1 \leq c_1$ $c_1 = 1$ için sağlanır

$$\boxed{f(n) \in \Theta(g(n))}$$

6) $f(n) \in ?(g(n))$

$$c_2.g(n) \leq f(n) \leq c_1.g(n)$$

$$c_2.n^{\frac{105}{100}} \leq n \log n \leq c_1.n^{\frac{105}{100}}$$

$c_2.\cancel{n}.n^{\frac{1}{20}} \leq \cancel{n}.\log n$

$c_2 \sqrt[20]{n} \leq \log n$

Polinom Fonksiyonlar log'dan hızlı büyür. Sonsuzda bu eşitlik sağlanamaz

$\cancel{n} \log n \leq c_1.\cancel{n}.\sqrt[20]{n}$

$\log n \leq c_1 \sqrt[20]{n}$

Polinom Fonksiyonlar log'dan hızlı büyür. Sonsuzda bu eşitlik sağlanır.

$$\boxed{f(n) \in O(g(n))}$$

7) $f(n) \in ?(g(n))$

$$c_2.g(n) \leq f(n) \leq c_1.g(n)$$

$$c_2.\log_2 n^3 \leq \log \sqrt{10n} \leq c_1.\log_2 n^3$$

$3c_2.\log_2 n \leq \frac{1}{2}\log 10n$

$c_2.\log n \leq \frac{\log 10}{6} + \frac{\log n}{6}$

$\left(c_2 - \frac{1}{6}\right)\log n \leq \frac{\log_2 10}{6}$

$c_2 = \frac{1}{12}$ $n = 2$

değerleri için eşitlik sağlanır.

$\frac{1}{2}\log 10n \leq 3c_1 \log_2 n$

$\frac{\log 10}{6} + \frac{\log n}{6} \leq c_1 \log n$

$\frac{\log 10}{6} \leq \left(c_1 - \frac{1}{6}\right)\log n$

$n = 2$ $c_1 = 5$ değerleri için eşitlik sağlanır.

$$\boxed{f(n) = \Theta(g(n))}$$

8) $f(n) \in ?(g(n))$

$$c_2.g(n) \leq f(n) \leq c_1.g(n)$$

$$c_2.(n+1)! \leq n! \leq c_1.(n+1)!$$

$c_2.(n+1).n! \leq n!$

$c_2.(n+1) \leq 1$

$n+1 \leq \frac{1}{c_2}$

$n \to \infty$ olduğundan c_2'nin bütün sonlu değerlerini geçecektir. Eşitsizlik sağlanamaz.

$n! \leq c_1.(n+1).n!$

$1 \leq c_1.(n+1)$

$\frac{1}{c_1} \leq n+1$

$n_0 = 1$ $c_1 = 1$ için eşitlik sağlanır

$$\boxed{f(n) = O(g(n))}$$

4- Big-Theta Çözümü ve İspatı

a. 1. soru: $f(n) = 2^{n+1} + 3^{n-1} = 2 \cdot 2^n + \frac{3^n}{3}$

n'in sonsuza gittiği düşünüldüğünde sabitler önemsiz hale gelir: $f(n) = 2^n + 3^n$. 3^n daha kapsayıcı olduğundan $f(n) \in \Theta(3^n)$ denilebilir.

İspat: $3^n \cdot c_2 \leq 2^n \cdot 2 + \frac{3^n}{3} \leq 3^n \cdot c_1$

$c_2 = \frac{1}{3}$ $n_0 = 1$ için sağlanır $\leftarrow \left(c_2 - \frac{1}{3}\right) \cdot 3^n \leq 2^n$

$2^n \cdot 2 + \frac{3^n}{3} \leq 3^n \cdot c_1$

$2^n \cdot 2 \leq \left(c_1 - \frac{1}{3}\right) 3^n$ $n_0 = 1$ $c_1 = 5$ için sağlanır

İki eşitsizlik de sağlandığından $\boxed{f(n) \in \Theta(3^n)}$ denilebilir.

b. 2. soru: $2n \log(n+2)^2 + (n+2)^2 \log\left(\frac{n}{2}\right)$

$$= 4n \log(n+2) + (n+2)^2 \left(\log n + \log \tfrac{1}{2}\right) = 4n \log(n+2) + (n+2)^2 \cdot (\log n - 1) = 4n \cdot \log(n+2) + (n^2 + 4n + 4)(\log n - 1)$$

bu ikisinin çarpımı ifadedeki en kapsayıcı terimdir: $f(n) \in \Theta(n^2 \log n)$

İspat:

$c_2 \cdot n^2 \log n \leq 2n \cdot \log(n+2)^2 + (n+2)^2 \cdot \log \frac{n}{2} \leq c_1 \cdot n^2 \log n$

$c_2 \cdot n^2 \log n \leq 4n \log(n+2) + (n+2) \cdot (\log n - 1) \leq c_1 \cdot n^2 \log n$

$n_0 = 2$ için $4c_2 \leq 16 + 0$ → $c_2 = 3$ değerleri için sağlanır

$n_0 = 2$ için $16 + 0 \leq c_1 \cdot 4 \Rightarrow c_1 = 5$ için sağlanır

İki eşitsizlik de sağlandığından $\boxed{f(n) \in \Theta(n^2 \log n)}$ denilebilir.

5- Big – Oh Asimptotik Notasyonu ile Yazma

$$\sum_{i=1}^{n} (i+1)\, 2^{i-1} = \sum_{i=1}^{n} (i+1)\, 2^i \cdot \frac{1}{2} = \frac{1}{2} \sum_{i=1}^{n} (i+1) \cdot 2^i = \frac{1}{2} \sum_{i=1}^{n} i \cdot 2^i + 2^i = \frac{1}{2}\left(\sum_{i=1}^{n} i\, 2^i + \sum_{i=1}^{n} 2^i\right)$$

$$= \frac{1}{2}\left(\left(2(2^n n - 2^n + 1)\right) + \sum_{i=1}^{n} 2^i\right) = \frac{1}{2}\left(\left(2(2^n n - 2^n + 1)\right) + (2(2^n - 1))\right) = \frac{1}{2}\left(2 \cdot 2^n \cdot n - \cancel{2 \cdot 2^n} + \cancel{2} + \cancel{2 \cdot 2^n} - \cancel{2}\right) = \cancel{\frac{1}{2}} \cdot \cancel{2}\, 2^n \cdot n$$

$O(2^n \cdot n)$

6- Backward Substitution (Soruda base case verilmediğinden $T(0) = 0$ olarak varsayılmıştır)

$T(n) = T(n-2) + 2n = T(n-4) + 4n - 4 = T(n-6) + 6n - 12$

$T(n-2) = T(n-4) + 2n - 4$

$T(n-4) = T(n-6) + 2n - 8$

$T(n) = T(n-2i) + 2in - 2i\left(\frac{2i}{2} - 1\right)$

$= T(n-2i) + 2in - 2i^2$

$= T(n-2i) + 2i(n-i)$

$n = 2i$ için

$= \underbrace{T(0)}_{0} + n\left(n - \frac{n}{2}\right)$

$= n^2 - \frac{n^2}{2} = \frac{n^2}{2}$ $\boxed{\in O(n^2)}$