Buble Sort

Öğrenci Adı: Emir Kerem ÖZTÜRK

Öğrenci Numarası: 20011613

Dersin Eğitmeni: M. Elif Karslıgil

Video Linki: -

Soru 1:

Algoritma Analizi Ödev-1

1) 'Master Theorem' yardımı ile çöz:

a) $T(n) = 9T(\frac{n}{4}) + n^2$

$a=9 \quad b=4 \quad d=2 \Rightarrow b^d = 16 \Rightarrow a < b^d$

$= T(n) \in O(n^2)$

b) $T(n) = 3T(\frac{n}{3}) + \log n$

$a=3 \quad b=3 \quad d=0 \Rightarrow b^d = 1 \Rightarrow a > b^d$

$= T(n) \in O(n^{\log_3 3})$

$= O(n)$

c) $T(n) = 3T(\frac{n}{2}) + n$

$a=3 \quad b=2 \quad d=1 \quad b^d = 2 \quad a > b^d$

$T(n) \in O(n^{\log_2 3})$

!Master Theorem kuralları

$a > b^d$ ise $T(n) = O(n^{\log_b a})$

$a = b^d$ ise $T(n) = O(n^d \log n)$

$a < b^d$ ise $T(n) = O(n^d)$

$T(n) = aT(\frac{n}{b}) + n^d$

Soru 2:

2) Big-Oh cinsinden karmaşıklıkları kodların altına yaz

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

Her zaman N defa döner $\boxed{O(N)}$

b)
```
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;
}
```

- dış loop N kere tekrar eder $\boxed{O(N^3)}$
- iç loop, her dış loop için $j=0$ dan $j<i$ kadar döner yani i defa ve aynı zamanda f1 çağırır j ile $(O(j))$

$= \sum_{i=1}^{N-1} \sum_{j=0}^{i=1} O(j) \Rightarrow \sum_{j=0}^{i=1} j = \frac{i(i-1)}{2} = O(i^2)$

$= \sum_{i=1}^{N-1} i^2 = \frac{N(N+1)(2N+1)}{6} \Rightarrow \frac{2N^3+3N^2+N}{6}$

$= N^3$

c)
```
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;
}
```

recurrence: $T(N) = N \cdot T(N-1) + O(1)$

$T(N) = N((N-1)T(N-2)+1)+1$

$= N(N-1)((N-2) \cdot T(N-3)+1)+N+1$

$T(N) = N! + N! + \ldots + 1$

$T(N) \in O(n!)$

d)
```
int f4(int N){
    if(N==0) return 0;
    return f4(N/2)+f1(N)+f1(N)+f1(N)+f4(N/2);
}
```

-recursive $2 * f4(\frac{N}{2}) + 3 * f1 \quad f1 = O(N)$

recurrence: $T(N) = 2T(\frac{N}{2}) + N$

Master Theorem: $a=2 \quad b=2 \quad d=1$

$b^d = 2 \quad a = b^d$

$T(N) \in O(N \log N)$

Soru 3:

3) Büyüme dereceleri için en uygun ifadeyi kullan ilişki yoksa X

Kurallar
$$\begin{cases} c_1 \cdot g(n) \le f(n) \le c_2 \cdot g(n) & \text{ise } \Theta \\ f(n) \le c \cdot g(n) & \text{ise } O \\ f(n) \ge c \cdot g(n) & \text{ise } \Omega \end{cases}$$

CEVAP	f(n)	g(n)
O	n^2	n^3
Ω	$n\lg n$	n
X	1	$3+\sin n$
Ω	3^n	2^n
O	4^{n+4}	2^{2n+2}
O	$n\lg n$	$n^{105/100}$
Θ	$\lg\sqrt{10n}$	$\lg n^3$
O	$n!$	$(n+1)!$

=> $f(n) \le c.g(n)$ n^3 sonsuza daha hızlı gider

=> $n\log n$ daha hızlı ∞'a gider

=> 1 sabit | $3+\sin(n)$ iki sayı arasında gider
NO RELATION

=> $3>2$ bundan dolayı $f(n)$ daha hızlı gider ∞

=> $g(n) = 2^n$ | $f(n) = n^4$ | $g(n)$ daha hızlı gider

=> $f(n) = n\log n$, $g(n) = n^{1.05}$ $\}$ $n\log n < n^{1.05}$ log

=> $f(n) = \log\sqrt{10n} = \frac{1}{2}\log(10n) \Rightarrow \frac{1}{2}\log 10 + \frac{1}{2}\log n$
$f(n) = \log n$
$g(n) = 3\log n$ $g(n) = f(n)$
growth rate to ∞

$f(n) = n!$ $g(n) = (n+1)!$ $\}$ a) 1 vs 2 b) 2 vs 6 c) 6 vs 24 d) 24 vs 120 ....

Soru 4:

4) Big-Theta cinsinden ifade edip çözümünü ispatla:

a) $2^{n+1} + 3^{n-1}$

$f(n) = 2.2^n + \frac{3^n}{3} \rightarrow 3^n > 2^n \rightarrow f(n) = \Theta(3^n)$

proof: $c_1 \cdot g(n) \le f(n) \le c_2 \cdot g(n)$ → has to be true

c_1 için

$c_1 . 3^n \le 2.2^n + \frac{3^n}{3}$ $\boxed{c_1 = \frac{1}{3}}$

$\Omega(3^n)$

c_2 için

$2.2^n + \frac{3^n}{3} \le c_2 . 3^n$ $\boxed{c_2 = 2}$

$O(3^n)$

b) $2n\lg(n+2)^2 + (n+2)^2 \lg\frac{n}{2}$

1) $2n.\log(n+2)^2 \Rightarrow \cancel{4}n.\lg(n\cancel{+2}) \Rightarrow \Theta(n\log n)$

2) $(n+2)^2 = n^2$, $\lg\frac{n}{2} \Rightarrow \lg n - \lg 2 \Rightarrow \Theta(n^2\log n)$ $\}$ $f(n) = \Theta(n^2\lg n)$

proof: $c_1.n^2\lg n \le 4n\lg n + n^2\lg n \le c_2.n^2\lg n$

c_1 için

$c_1.n^2\lg n \le \cancel{4n\lg n} + n^2\lg n$ $\boxed{c_1 = 1}$

$1.n^2\lg n \le 1.n^2\lg n$ ✓

c_2 için

$\cancel{4n\lg n} + n^2\lg n \le c_2.n^2\lg n$ $\boxed{c_2 = 2}$

$1n^2\lg n \le 2n^2\lg n$ ✓

Soru 5:

5) Toplam ifadesinin Büyüme derecesini hesapla Big-Oh asimptotik notasyonunu kullanarak yazı

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

$$= \sum_{i=1}^{n} i.2^{i-1} + \sum_{i=1}^{n} 2^{i-1} \nearrow 2^0+2^1+2^2 \ldots + 2^{n-1} = \frac{2^n-1}{2-1} = 2^n-1$$

$$\sum_{i=1}^{n} i*2^{i-1} + \sum_{i=1}^{n} 2^{i-1} \Rightarrow n.2^n + \cancel{2^n} \rightarrow \text{Big-Oh } O(n2^n)$$

Soru 6:

6) $T(n) = T(n-2)+2n$ 'backword sub' ile çöz

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

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

$$T(n) = T(n-2i)+2n+2(n-2)+2(n-4) \ldots +2(n-2(i-1))$$

$n-2 \leq 0$ olduğunda biter $\boxed{i=\frac{n}{2}}$

$$\sum_{j=0}^{i-1}(n-2j) \rightarrow \boxed{T(n) \in \Theta(n^2)}$$

seri aritmetiktir

$$S = i(n-i+1)$$

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

$T(1)$ veya $n-2i=0$

$T(0)$ için

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

Karşılaşılan Sorunlar:

El ile yazılı olan bir ödev olduğundan dolayısıyla eski ders notlarına ve uygulamada yaptığımız örneklere bakıp anlayamadığım bir bölüm olmadı. Sorular arasından en uzun süreyi 2. Soruda kodların zaman karmaşıklıklarını Big-Oh cinsinden yazarken, içerisinde recursive olan soruda harcadım.