Gebze Technical University

Computer Engineering

CSE - 321

Introduction to Algorithm Design

Homework - I

Yunus ÇEVİK

141044080

Course Teacher: Didem GÖZÜPEK

Course Asistance: Mehmet Burak KOCA

1-) Answer in detail the question that are shown blown by using asymtotic notation, yes / no answers and plagiarisation from the web will not be accepted.

a) Explain why the statement, "The running time of algorithm A is at least $O(n^2)$," is meaningless.

b) Are the following true?

i) $2^{n+1} = O(2^n)$?

ii) $2^{2n} = O(2^n)$?

c) Are the function below polynomially bounded?

i) $[\log n]!$

ii) $[\log \log n]!$

Answer -1

a) A algoritmasının çalışma zamanı en az $O(n^2)$ ifadesinin anlamsız olmasının nedeni en az (at least) denilmesinden dolayıdır. Big-Oh notasyonu o fonksiyonun "upper bound" olduğunu belirtir. Bu yüzden en az (at least) yerine en çok (at most) kullanılmış olsaydı, bu ifade doğru bir şekilde aktarılmış olacaktı. Diğer bir değişle en az (at least) kullanılacaksada $O(n^2)$ Big-oh notasyonu yerine $\Omega(n^2)$ Omega notasyonu tercih edilmeliydi.

b) i) $2^{n+1} = O(2^n)$ → c değeri "≤" ifadesini sağlaması için yazılmış pozitif bir değerdir.

$T(N) \leq c \cdot f(N)$ when $N \geq n_0$

$\cancel{2^n} \cdot 2^1 \leq c \cdot \cancel{2^n} \Rightarrow 2 \leq c$, $n \geq 1$ olduğu durumda

$2^{n+1} \in O(2^n)$

ii) $2^{2n} = O(2^n)$

$T(N) \leq c \cdot f(N)$ when $N \geq n_0$

$\cancel{2^n} \cdot 2^n \leq c \cdot \cancel{2^n} \Rightarrow 2^n \leq c$ c değeri pozitif bir sabit sayı olduğu için N değeri sınırsızdır. Tüm $N \geq n_0$ ve c değeri doğru olmaz

$2^{2n} \notin O(2^n)$

Answer-1

c) i) Eğer polinomial bounded tanımlanmalı istenirse:

$$f(n) \leq c\, n^k$$

Eşitliğin her iki tarafı logaritma eklenmelidir.

$$\log f(n) \leq \log c + \log n$$

Böylece, Eğer $\log f(n) = \Theta(\log n)$ olduğunda fonksiyon polinomial bounded olur.

Eğer $m = [\log n]$

$$\log m! = \Theta(m \log m) = \Theta([\log n] \log [\log n])$$

Böylece, $[\log n]!$ polinomial bounded değildir.

ii) Eğer $p = [\log(\log n)]$:

$$\log p! = \Theta(p \log p) = \Theta([\log(\log n)] \log [\log(\log(\log n))]) =$$
$$= \Theta(\log(\log n) \log(\log(\log n)))$$
$$= o(\log(\log n) \log(\log n))$$
$$= o(\log^2(\log(\log n)))$$
$$= o(\log n)$$

Yukarıda belirtilen son adımda, her poligoritmik fonksiyon her pozitif polinomial fonksiyondan daha yavaş bir şekilde büyüdüğü sonucu çıkarılabilir.

$$\log^2(\log n) = o(\log n)$$

Bu durumda, $\log([\log(\log n)]!) = O(\log n)$, böylece $[\log(\log n)]!$ ifadesi polinomial bounded'tır.

2-) In each of the following situations, indicate whether $f \in O(g)$, or $f \in \Omega(g)$, or both (in which case $f \in \Theta(g)$).

	$f(n)$	$g(n)$
a)	$n^2/\log n$	$n(\log n)^2$
b)	$(\log n)^{\log n}$	$n/\log n$
c)	$\sqrt{n}$	$(\log n)3$
d)	$(\log n)^{\log n}$	$2^{(\log n)^\wedge 2}$
e)	$\sum_{i=1}^{n} i^k$	n^{k+1}
f)	$100n + \log n$	$n + (\log n)^2$
g)	$n^{1/2}$	$5^{\log_2 n}$
h)	$n^{0.1}$	$(\log n)^{10}$

Answer-2

$$\lim_{n\to\infty} \frac{f(N)}{g(N)} = \begin{cases} =0 \Rightarrow f(N) = O(g(N)) \\ =c\neq 0 \Rightarrow f(N) = \Theta(g(N)) \\ =\infty \Rightarrow f(N) = \Omega(g(N)) \end{cases}$$

a) $$\lim_{n\to\infty} \frac{\frac{n^2}{\log n}}{n(\log n)^2} = \lim_{n\to\infty} \frac{n^{\cancel{2}}}{\cancel{n}(\log n)^3} = \frac{n}{(\log n)^3} = \frac{\infty}{\infty}$$

L'hospital $$\lim_{n\to\infty} \frac{1}{3.(\log n)^2 \cdot \frac{1}{\ln 2 . n}} = \lim_{n\to\infty} \frac{\ln 2 . n}{3.(\log n)^2} = \frac{\infty}{\infty}$$

L'hospital $$\lim_{n\to\infty} \frac{\ln 2}{6(\log n) \cdot \frac{1}{\ln 2 \cdot n}} = \lim_{n\to\infty} \frac{(\ln 2)^2 . n}{6(\log n)} = \frac{\infty}{\infty}$$

L'hospital $$\lim_{n\to\infty} \frac{(\ln 2)^2}{6 \cdot \frac{1}{\ln 2 \cdot n}} = \lim_{n\to\infty} \frac{(\ln 2)^3 . n}{6} = \frac{\infty}{6} = \infty$$

$f(N) \in \Omega(n(\log n)^2)$

Answer2

c) $\lim_{n\to\infty} \frac{\sqrt{n}}{(\log n).3} \Rightarrow \lim_{n\to\infty} \frac{n^{1/2}}{(\log n)^2}$

L'hospital $\frac{\frac{1}{2}\cdot n^{-1/2}}{3.\frac{1}{(\ln 2)n}} \Rightarrow \frac{(\ln 2)n}{6}\cdot n^{-1/2} = \frac{\ln 2\, n^{1/2}}{6}$

$\lim_{n\to\infty} \frac{\ln 2\sqrt{n}}{6} = \frac{\infty}{6} = \infty$ $\quad f(N) \in (3.\log n)$

e) $\lim_{n\to\infty} \frac{1^k+2^k+\dots+(n-1)^k+n^k}{n^k+1}$ $\Rightarrow$ n^k değeri diğer değerlere göre daha baskındır ve türevini alacağımız zaman bu değerden daha küçüklerin önemi yoktur.

$\lim_{n\to\infty} \frac{n^k}{n^k.n^1} = \lim_{n\to\infty} \frac{1}{n} = \frac{1}{\infty} = 0$ $\quad f(N) \in O(n^{k+1})$

b) $\lim_{n\to\infty} \frac{(\log n)^{\log n}}{n/\log n} \Rightarrow \lim_{n\to\infty} \frac{(\log n)^{\log n+1}}{n} = \frac{\infty}{\infty}$ $\rightarrow$ Bu ifadenin türevi çok uzun olduğu için L'hospitalde payda kısmındaki değer 1 çıktığı için üstte yer alan yer "∞" dur.

L'hospital $\lim_{n\to\infty} \frac{((\log n)^{\log n+1})'}{1}$

$= \frac{\infty}{1} = \infty$ $\quad f(N) \in \Omega(n/\log n)$

f) $\lim_{n\to\infty} \frac{100n+\log n}{n+(\log n)^2} = \frac{\infty}{\infty}$

L'hospital $\lim_{n\to\infty} \frac{100+\frac{1}{(\ln 2).n}}{1+2.\log n.\frac{1}{(\ln 2)n}} = \frac{100(\ln 2).n+1}{(\ln 2).n} \cdot \frac{(\ln 2).n}{(\ln 2).n+2\log n}$

$\lim_{n\to\infty} \frac{100(\ln 2).n+1}{(\ln 2).n+2\log n} = \frac{\infty}{\infty}$

L'hospital $\lim_{n\to\infty} \frac{100(\ln 2)}{\ln 2+\frac{2}{\ln 2.n}} \Rightarrow \frac{100(\ln 2)}{\frac{(\ln 2)^2 n+2}{(\ln 2)n}} = \frac{100(\ln 2).(\ln 2).n}{(\ln 2)^2 n+2} = \frac{\infty}{\infty}$

L'hospital $\lim_{n\to\infty} \frac{100(\ln 2)^2}{(\ln 2)^2} = 100$ sonuç sabit değerdir.

$f(N) \in \Theta(n+(\log n)^2)$

Answer-2

g) $\lim_{n\to\infty} \frac{n^{1/2}}{5^{\log_2 n}} = \lim_{n\to\infty} \frac{n^{1/2}}{n^{\log_2 5}} = \lim_{n\to\infty} \frac{n^{0,5}}{n^{2,32}}$

$\lim_{n\to\infty} n^{0,5-2,32} = \lim_{n\to\infty} \frac{1}{n^{1,82}} = \frac{1}{\infty} = 0 \quad f(N) \in O(5^{\log_2 n})$

h) $\lim_{n\to\infty} \frac{n^{0,1}}{(\log n)^{10}} = \frac{\infty}{\infty}$

L'hospital $\lim_{n\to\infty} \frac{0,1 \cdot n^{-0,9}}{10 \cdot (\log n)^{9} \cdot \frac{1}{(\ln 2) n}} = \boxed{\frac{0,1 \cdot (\ln 2) \cdot n^{0,1}}{10 (\log n)^{9}}} = \frac{\infty}{\infty}$

L'hospital $\lim_{n\to\infty} \frac{(0,1)^2 \cdot (\ln 2)\, n^{-0,9}}{90 \cdot (\log n)^{8} \cdot \frac{1}{(\ln 2) n}} = \frac{(0,1)^2 (\ln 2)^2 \cdot n^{0,1}}{90 \cdot (\log n)^{8}} = \frac{\infty}{\infty}$

Yukarıda tekrarlı bir şekil olduğu için "x" ile gösterilen yer 1 den 10 kadar değişen tekrar değeridir.

1 den 10 kadar artıyor

$\frac{(0,1)^{x} \cdot (\ln 2)^{x} \cdot n^{0,1}}{x! \cdot 1} = \frac{(0,1)^{10} \cdot (\ln 2)^{10} \cdot n^{0,1}}{10! \cdot 1}$

faktoriyel bir şekilde 10 dan 1 kadar azalıyor

$\lim_{n\to\infty} \frac{(0,1)^{10} \cdot (\ln 2)^{10} \cdot n^{0,1}}{10!} = \frac{\infty}{10!} = \infty \quad f(N) \in \Omega((\log n)^{10})$

d) $\lim_{n\to\infty} \frac{(\log n)^{\log n}}{2^{(\log_2 n)^2}} = \frac{\infty}{\infty}$ L'hospital $\lim_{n\to\infty} \frac{\frac{2 \ln(n)}{\ln^2(2) \cdot n}}{\frac{2 \cdot 2^{\log_2 n^2} \cdot \log_2 n}{n}} = \frac{\ln(n)}{\ln^2(2) \cdot 2^{\log_2 n^2} \cdot \log_2 n} = \frac{\infty}{\infty}$

L'hospital $\lim_{n\to\infty} \frac{1/n}{(\ln 2) \cdot (n + 2n \cdot \ln(n))} = \frac{0}{\infty}$

L'hospital $\lim_{n\to\infty} \frac{-1/n^2}{2 \ln(n) + 3} = \frac{0}{\infty}$ L'hospital $\lim_{n\to\infty} \frac{2/n^3}{2/n} = \frac{\cancel{2}}{n^3} \cdot \frac{n}{\cancel{2}} = \frac{1}{n^2}$

$\lim_{n\to\infty} \frac{1}{n^2} = \frac{1}{\infty} = 0 \quad f(N) \in O(2^{(\log_2 n)^2})$

3-) For each of the following functions, indicate the function's growth rate when its input is increased by one ($n \to n+1$) and order the functions according to their growth rates.

$\log_2 n, \sqrt{n}, n!, n^2, n, 2^n, 3^n$

Answer-3

Bu soruda $n \to (n+1)$ bağıntısına bakarak büyüme oranını bulmak için n değerine rastgele bir değer verip oranları bulup sıralaya biliriz.

$n = 32$ olduğunu varsayalım

$\log_2 n \Rightarrow \log_2 32 = 5$
$\log_2 n+1 \Rightarrow \log_2 33 \cong 5{,}044$
$0{,}044^{+}$ artmıştır

Artış oranı $= \frac{\log_2 n+1}{\log_2 n} = \frac{5{,}044}{5} = 1{,}0088$

$\sqrt{n} \Rightarrow \sqrt{32} \cong 5{,}66$
$\sqrt{n+1} \Rightarrow \sqrt{33} \cong 5{,}74$
$0{,}08^{+}$ artmıştır

Artış oranı $= \frac{\sqrt{33}}{\sqrt{32}} \cong 1{,}014 \Leftarrow \frac{\sqrt{n+1}}{\sqrt{n}}$

$n! \Rightarrow 32!$
$(n+1)! \Rightarrow 33!$
33^{*} kat artmıştır

Artış oranı $= \frac{33!}{32!} = \frac{33 \cdot \cancel{32!}}{\cancel{32!}} = 33 \Leftarrow \frac{(n+1)!}{n!}$

$n^2 \Rightarrow (32)^2$
$(n+1)^2 \Rightarrow (33)^2$
65^{+} artmıştır

Artış oranı $= \frac{(33)^2}{(32)^2} \cong 1{,}063 \Leftarrow \frac{(n+1)^2}{n^2}$

$n \Rightarrow 32$
$n+1 \Rightarrow 33$
1^{+} artmıştır

Artış oranı $= \frac{33}{32} \cong 1{,}031 \Leftarrow \frac{n+1}{n}$

$2^n \Rightarrow 2^{32}$
$2^{n+1} \Rightarrow 2^{33}$
2^{*} artmıştır

Artış oranı $= \frac{2^{33}}{2^{32}} = 2 \Leftarrow \frac{2^{n+1}}{2^n}$

$3^n \Rightarrow 3^{32}$
$3^{n+1} \Rightarrow 3^{33}$
3^{*} artmıştır

Artış oranı $= \frac{3^{33}}{3^{32}} = 3 \Leftarrow \frac{3^{n+1}}{3}$

$$\log_2 n < \sqrt{n} < n < n^2 < 2^n < 3^n < n!$$

Fonksiyonların artış oranlarına göre sıralanmış hali üst tarafta belirtilmiştir.

4-) Analyze the complexity in time (big-Oh notation) of the following operations at a given binary search tree (BST) that has height n:

a) FindMax.

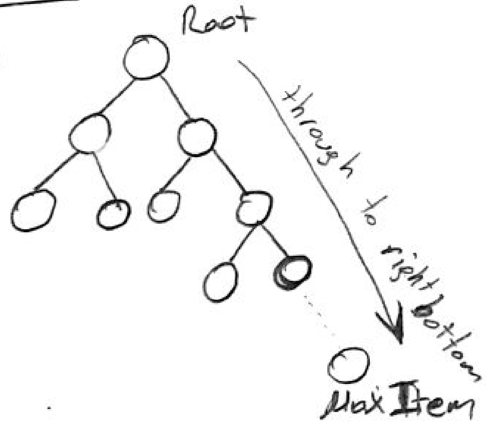
b) Insert a new element

c) Delete a leaf node

d) Printing BST using the in order traversal

Answer-4

a)



Bir Binary Search Tree'de en büyük eleman araması yapılırken "Root"dan başlanır, Root'un "Right-Child"larına bakarak bir döngü eşliğinde ilerlenir. Tree'nin en sağ alt kısımda bulunan Node değeri, ağacın en büyük elemanıdır.

En büyük eleman aranırken sağ çocukların kaçtane olduğun bilmediğimizden arama işlemi n kadar ilerler. Bu nedenle FindMax metodunun complexity analizi döngünün sıfırdan başlayıp n kadar gitmesinden dolayı

$T(N) = O(n)$ Linear Time kadardır.

b) Binary Search Tree'de yeni bir eleman eklenmesi için yeni elemanın değeri "Root" ile karşılaştırılır. Yeni eleman Root'tan küçük ise Root'un soluna, büyük ise Root'un sağına eklenir. Ancak Root'un sağ ve sol çocukları dolu ise yeni değerin çocukların değerlerine göre büyük-küçük karşılaştırılmaları sonucu ağaçta bir yere yerleştirilir. Ekleme işlemi, arama işlemi ile aynıdır. Bu yüzden ekleme yapılırken Binary Search Tree'de ekleme işleminin complexity analizi $T(N) = O(n)$ Linear Time kadar sürmektedir.

Answer - 4

c) Bir önceki aşamalarda belirtilen bir çok basamak Binary Search Tree'de silme işlemi yapılacağı zamanda gerçekleştirilir. Silme işlemindeki en önemli unsur arama yapmaktır. Sileceğimiz eleman ağaç içinde aranır ve daha sonra silme işlemi gerçekleştirilir. Arama işleminin complexity analizi $T(N) = O(n)$ olduğu için silme işleminde de $T(N) = O(n)$ Linear zaman kadar sürmektedir. Ancak bu işlem worst-case durumu için geçerlidir. Best-case durumunda silinecek elemanın "Root" olduğunu varsayarsak $T(N) = O(1)$ constant zaman kadar sürmektedir.

d) Binary Search Tree'de Node değerlerini print etmek için tüm Node ları gezmemiz gerekir. Traversal yöntemlerinden "Pre Order", "Post Order" ve "In Order" Traversal yöntemlerinden "In Order Traversal" ele alacak olursak n kadarlık bir ağacı ekrana bastırmak için Find metodundaki gibi $T(N) = O(n)$ lineer zamanda gezerek ekrana print işlemi uygulamaktadır.

5-) Find the complexity in time (big-Oh notation) of the following program.

a)

```
void function (int n)
{
    int count = 0;
    for(int i=n/2; i<=n; i++)
        for(int j=1; j<=n; j=2*j)
            for(int k=1; k<=j; k=k*2)
                count++;
}
```

Answer-5

a) → 1. For döngüsü n/2 değerinden başlayıp n'e kadar birer birer artarak dönmektedir. $T(n) = n/2$ kadar dönme işlemi yapar.

→ 2. For döngüsü 1 değerinden başlayıp n'e kadar 2 şer kat 2 şer kat artarak dönmektedir. $T(n) = \log_2 n$ kadar dönme işlemi yapar.

→ 3. For döngüsü 1 değerinden başlayıp j'e kadar 2 ser kat 2 ser kat artarak dönmektedir. $T(n) = \log_2 n$ kadar dönme işlemi yapar.

Not: 2 ve 3 For döngüsünde exponential olarak artış vardır.

$(1\ 2\ 4\ 8\ 16 \ldots\ldots 2^n)$

$$\left.\begin{aligned} T_1(N) &= n/2 \\ T_2(N) &= \log n \\ T_3(N) &= \log n \end{aligned}\right\} \quad \begin{aligned} T(N) &= T_1(N) \cdot T_2(N) \cdot T_3(N) \\ T(N) &= \frac{n \cdot \log n \cdot \log n}{2} \end{aligned} \qquad T(N) = n \cdot (\log n)^2$$

$$\boxed{\begin{aligned} T(N) &\in O(n \cdot (\log n)^2) \\ T(N) &\in O(n \cdot \log^2 n) \end{aligned}}$$