COM3067 – Algorithms

2020-Fall

Homework1 Solutions

Res. Asst. Zeynep YILDIRIM
E-mail: yildirimz@ankara.edu.tr

Homework1

Question-1

Order the following functions according to their order of growth (from the lowest to the highest). If any two or more are of same order, indicate which.

$$f_{12}(n) < f_{14}(n) < f_{11}(n) < f_{10}(n) < f_2(n) < f_{13}(n) < f_3(n) < f_4(n) < f_1(n) < f_8(n) < f_9(n) < f_7(n) < f_5(n) < f_6(n)$$

$f_1(n) = n^2 + logn$

$f_2(n) = \sqrt{n}$

$f_3(n) = n - 1000$

$f_4(n) = nlogn$

$f_5(n) = 2^n + n^{10}$

$f_6(n) = n^5 + 3^n$

$f_7(n) = n^{11} . 2^{2logn}$

$f_8(n) = n^{12} + n^{10}$

$f_9(n) = n^{12} . logn$

$f_{10}(n) = n^{1/3} + logn$

$f_{11}(n) = (logn)^2$

$f_{12}(n) = 10^{15}$

$f_{13}(n) = \frac{n}{logn}$

$f_{14}(n) = loglogn$

Homework1
Question-2

What value is returned by the following algorithm? $\frac{n^3-n}{3}$

What is its basic operation? How many times is the basic operation executed? $r \leftarrow r + 1$

Give the worst-case running time of the algorithm using Big Oh notation. $\mathrm{T(n)} = O(n^3)$

MISSISSIPPI (n)
input : *an integer n*
$r \leftarrow 0$
for i = 1 to n
    for j = i + 1 to n
        for k = i + j – 1 to n
            $r \leftarrow r + 1$
return r

n
n-i-1
n-i-j+1

$$\sum_{i=1}^{n} \sum_{j=i+1}^{n} \sum_{k=i+j-1}^{n} 1$$

Homework1

Question-2

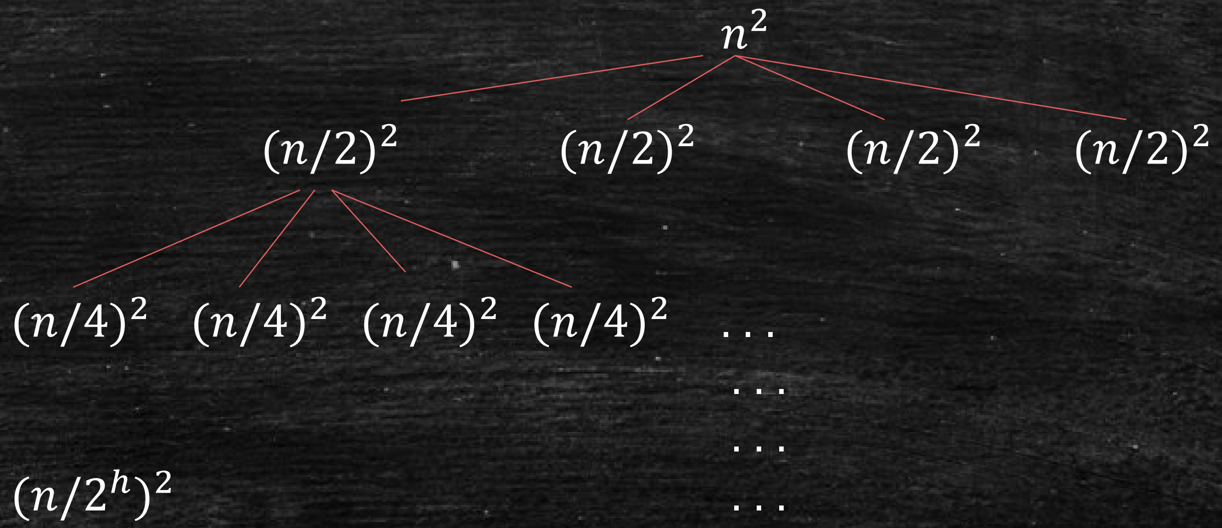
$$\sum_{i=1}^{n}\sum_{j=i+1}^{n}\sum_{k=i+j-1}^{n} 1 = \sum_{i=1}^{n}\sum_{j=i+1}^{n}(n-i-j+2) \Rightarrow \sum_{i=1}^{n}\sum_{j=i+1}^{n}(n-i+2) - \sum_{i=1}^{n}\sum_{j=i+1}^{n} j \Rightarrow$$

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

Homework1
Question-3

Solve the following recurrence relation using recursion tree method.

$$T(n) = \begin{cases} 1 & , if\ n \leq 1 \\ 4T(n/2) + n^2 & , if\ n > 1 \end{cases}$$



* $(n/2^h)^2 \Rightarrow$ h is depth of the tree.

Homework1

Question-3

$$T(n) = \begin{cases} 1 & , if\ n \leq 1 \\ 4T(n/2) + n^2, & if\ n > 1 \end{cases}$$

$n/2^h = 1$ => $h = \log_2 n = \lg n$

Sum to levels of tree:

$T(n) = n^2 + 4(n/2)^2 + 4^2(n/4)^2 + ... + 4^h(n/2^h)^2$

$T(n) = n^2 + 4(n/2)^2[1 + 4/4 + (4/4)^2 + ... + (4/4)^{h-1}]$

$T(n) = n^2 + (\frac{4n^2}{4})[h - 1]$

$T(n) = hn^2 = logn.n^2 = O(n^2.logn)$

Homework1

Question-4

Solve the following recurrence relation using recursion tree method.

$$T(n) = \begin{cases} 1 & ,if\ n \leq 1 \\ T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n, & if\ n < 1 \end{cases}$$

$$n$$

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

$$\left(\frac{n}{9}\right) + \left(\frac{2n}{9}\right) + \left(\frac{2n}{9}\right) + \left(\frac{4n}{9}\right) = n$$

$$\cdots$$

$$\cdots$$

$$\cdots$$

* $(n/(3/2)^h) \Rightarrow$ h is depth of the tree.

Homework1

Question-3

$$T(n) = \begin{cases} 1 & ,if\ n \le 1 \\ T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n, if\ n < 1 \end{cases}$$

$n/(3/2)^h = 1$ => $h = \log_{3/2} n$

Sum to levels of tree:

$T(n) = h.n$

$T(n) = h.n = \log_{3/2} n\,.n = O(n.logn)$

Homework1
Question-5

What value is returned by the following algorithm? $\frac{n^2-n}{2}$

What is its basic operation? How many times is the basic operation executed? **if** $a_{ij} \neq a_{ji}$

Give the worst-case running time of the algorithm using Big Oh notation. $\mathrm{T(n)} = O(n^2)$

Missouri ($A = (a_{ij})_{\boldsymbol{nxn}}$)
input : *an nxn matrix of real numbers*
$r \leftarrow 0$
for i = 1 to n-1 → n
  for j = i + 1 to n → n-i-1
    if $a_{ij} \neq a_{ji}$
      return false
return true

$$\left.\begin{matrix} n \\ n-i-1 \end{matrix}\right\} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} 1$$

Homework1

Question-5

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} 1 = \sum_{i=1}^{n-1} (n - (i + 1) + 1) \Rightarrow$$

$$= \sum_{i=1}^{n} (n - i) = (n - 1) + (n - 2) + \cdots + \big(n - (n - 1)\big) \Rightarrow$$

$$= \frac{(n - 1).\,n}{2}$$

Homework1
Question-6

Solve the following recurrence relation using Master Theorem.

$$\mathrm{T(n)} = \begin{cases} 1 & , if\ n \leq 2 \\ 2T\left(\frac{n}{2}\right) + n.logn, & if\ n > 2 \end{cases}$$

$$a = 2, b = 2 \ => \ p \ = \ \log_2 2 = 1$$

$$f(n) = n.logn \quad => \ n^p = n, k = 1 \quad => Case\ 2$$

$$T(n) = \Theta(n \log^2 n)$$

Homework1
Question-7

Solve the following recurrence relation using Master Theorem.

$$\mathrm{T(n)} = \begin{cases} 1 & , if\ n \le 2 \\ 3T\left(\frac{n}{3}\right) + \sqrt{n}, & if\ n > 2 \end{cases}$$

$$a = 3, b = 3 \ => \ p \ = \ \log_3 3 = 1$$

$$f(n) = \sqrt{n} => \ \ n^p > n^{1/2}, \ => Case\ 1$$

$$T(n) = \Theta(n)$$

Homework1
Question-8

What does the following recursive algorithm compute? Given a sequence of integers, returns the min of the sequence.

Set up a recurrence relation for the running time of the algorithm and solve it using backward substitution.

RioGrande $(\langle \boldsymbol{a_i}, \boldsymbol{a_{i+1}}, \dots, \boldsymbol{a_j} \rangle)$
input : *a sequence of integers*
if $i = j$
  return a_i
else
  mid ← $(i + j)/2$
  temp1 ← RioGrande$(\langle \boldsymbol{a_i}, \dots, \boldsymbol{a_{mid}} \rangle)$
  temp2 ← RioGrande$(\langle \boldsymbol{a_{mid}}, \dots, \boldsymbol{a_j} \rangle)$
  if $temp1 \le temp2$
    return temp1
  else
    return temp2

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

It doesn't matter if this value is 1, 2, 3, ... It is ineffective in calculations because it is a constant number.

Homework1

Question-8

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

$$T(n) = 2.2T\left(\frac{n}{4}\right) + 2 + 1 = 2^2T\left(\frac{n}{4}\right) + 2 + 1 \Rightarrow$$

$$T(n) = 2^2.2T\left(\frac{n}{8}\right) + 4 + 2 + 1 \Rightarrow$$

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

$$T(1) = 1, \qquad \frac{n}{2^i} = 1, \qquad i = logn$$

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

$$T(n) = O(n)$$

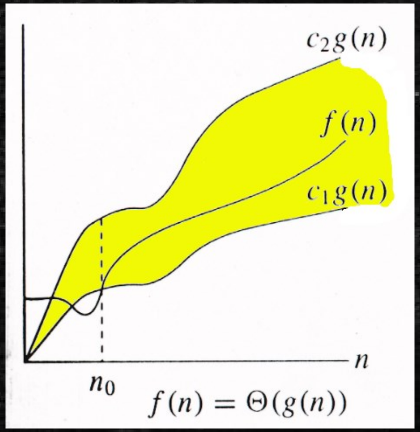
Homework1
Question-9

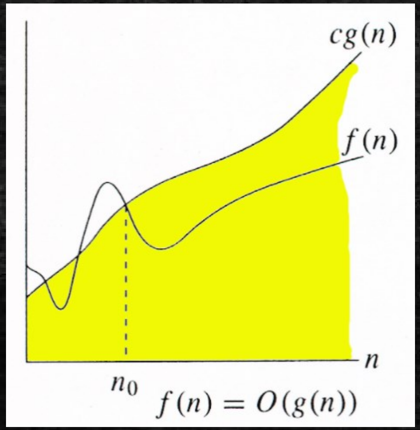
For each of the following pair of functions, either $f(n)$ is $O(n)$, $f(n)$ is $\Omega(n)$, or $f(n)$ is $\Theta(n)$. Determine which relationship is correct and briefly explain why.

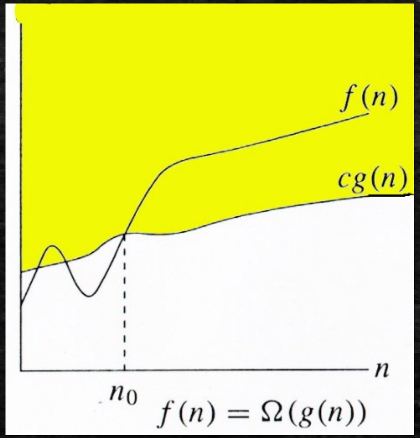
a) $f(n) = \sqrt{n}, g(n) = \log n^2$ → $\Omega(g(n)) \Rightarrow \sqrt{n} > log n^2$

b) $f(n) = n, g(n) = log^2 n$ → $\Omega(g(n)) \Rightarrow n > log^2 n$

c) $f(n) = 2^n, g(n) = 3^n$ → $O(g(n)) \Rightarrow 2^n < 3^n$







Thanks☺