P-1

Date - 25-09-2021

Name - Amit A. Pandit
email - apandit1@binghamton.edu
Bnumber - B00912493

CS-575 - Design & Analysis of Algorithms
Home Work - II

sol^n 1) → let Consider
The logs
$\lg^2 n$, $n\log n$, $\ln\ln n$, $\ln n$

Order of log's -
$\ln\ln n \leq \ln n < n\log n < \lg^2 n$

Now considering
$(n+1)!$ and $n!$
$(n+1)! > n!$

following function which are equal to each other

i) $4^{\lg n} = 2^{2\log n}$
$= 2^{\log n^2}$
$= n^2$

ii) $2^{\lg n} = n^{\log 2}$
$= n$

iii) $n^{1/\lg n}$ by taking log
$= \log n^{1/\lg n}$
$= \frac{1}{\lg n} \lg n$

iv) Now e^n and 2^n
value of $e \approx 2.718$
Thus $2^n < e^n$

P-2

v)

Consider $\lg n!$ & $(\lg n)^{\lg n}$

let $\lg n = x$

Thus $(\lg n)! = x!$ & $\lg n^{\lg n} = x^x$

Thus $x! < x^x$

Therefore $(\lg n)! < \lg n^{\lg n}$

The order is :-

$2^{2^{n+1}} > 2^{2n} > (n+1)! > n! > e^n >$

$n2^n > 2^n > (\frac{3}{2})^n > \frac{n^{\lg\lg n}}{(\lg n)^{(\lg n)}},$ $> (\lg n)! > n^3$

$> \{\lg n, n^2 > n\lg n, \lg(n!)\} > n, 2^{\lg n} > (\sqrt{2})^{\lg n} >$

$2^{\sqrt{2}\lg n} > \lg^2 n > \ln n > \sqrt{\lg n} > \ln\ln n >$

$2^{\lg^* n} > \cancel{\lg(\lg^* n)}, \lg^* n > \lg(\lg^* n) > {}^1/_n\ {}^1/_{\lg n}$

$\lg^*(\lg n)$

P-3

Q2) sol^n →

$T(n) = aT(n/b) + f(n)$ —— master theorm —— (1)

a) $T(n) = 2T(n/2) + n^3$ —— given eq^n

$a = 2 \quad b = 2 \quad f(n) = n^3$

$n^{\log_b^a} = n^{\log_2^2} = n^1 = n < n^3$

$f(n) = \Omega(n^{\log_b a + \epsilon})$ —— 3rd case of Master Theorm

$af(n/b) = 2f(n/2)$

$= 2(n/2)^3$

$= n^3/4$

$n^3/4 \leq cn^3$

$\Rightarrow 1/4 \leq c$

$\Rightarrow c = 0.25$

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

$\boxed{T(n) = \Theta(n^3)}$

b) $T(n) = T(9n/10) + n$ —— given eq^n

$a = 1 \quad b = 9/10 \quad f(n) = n$ —— from (1)

$n^{\log_b^a} = n^{\log_{9/10}^1} = n^0 = 1 = \Theta(1)$ — 2nd case of Master theorm

$\Rightarrow \boxed{T(n) = \Theta(\log n)}$

c) $T(n) = 16T(n/4) + n^2$ — given

$a=16 \quad b=1/4 \quad f(n)=n^2$ — from ①

$n^{\log_b a} = n^{\log_{1/4} 16} = n^{\log_4 4^2} = n^2 = \Theta(n)^2$

— 2^{nd} case

$T(n) = \Theta(n^{\log_3 9} \log n)$

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

d) $T(n) = 7T(n/3) + n^2$

$a=7 \quad b=3 \quad f(n)=n^2$ — fr①

$n^{\log_b a} = n^{\log_3 7} = n^{1.77} < n^2$ — 3^{rd} case

$af(n/b) = 7f(n/3) = 7(n^2/9) = \frac{7n^2}{9}$

$\frac{7n^2}{9} \leq cn^2 \Rightarrow 7/9$ is < 1

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

e) $T(n) = 7T(n/2) + n^2$

$a=7, \quad b=2, \quad f=n^2$

$n^{\log_b a} = n^{\log_2 7} = n^{2.8} > n^2$ — 1^{st} case

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

P-5

f) $T(n) = 2T(n/4) + n^{1/2}$

$a = 2 \quad b = 4 \quad f(n) = n^{1/2}$

$n^{\log_b a} = n^{\log_4 2} = n^{0.5}$

$= \Theta(n^{0.5})$ — 2^{nd} Case

$T(n) = \Theta(n^{\log_a b} \log n)$

$$\boxed{T(n) = \Theta(n^{0.5} \log n)}$$

g) $T(n) = T(n-1) + n$

by using recursion method

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

$T(n-2) = T(n-3) + (n-2) \ldots \therefore T(n-(k+1)) + (n-k)$

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

Assume $n-(k+1) = 1$

$k = n-2$

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

i.e. largest nterm is n^2

$$\boxed{\text{Complexity} = \Theta(n^2)}$$

h) $T(n) = T(n^{1/2}) + 1$

let $n = 2^x$ therefore $x = \log_2 n$

$T(n) = T(n^{1/2}) + 1$

$T(2^x) = T(2^{x/2}) + 1$

after reduced to

$T(x) = T(x/2) + 1$

$a = 1 \quad b = 2 \quad f(n) = 1$ — from (1)

$n^{\log_b a} = n^{\log_2 1} = n^0 = 1$ — (2^{nd} case)

$T(n) = \Theta(n^{\log_b a} \log n)$

$\boxed{T(n) = \Theta(\log_2 n)}$

Q 3) Sol^n

- a)

		Cost	time
1)	for i = 1 to n	C_1	$(n+1)$
2)	k[i] = 0	C_2	n
3)	for i = 1 to n	C_3	$(n+1)$
4)	for j = 1 to n	C_4	$\left[\frac{(n)(n+1)}{2}\right] + n$
5)	k[i] = k[i] + j	C_5	$n.(n+1)/2$

$T(n) = C_1(n+1) + C_2(n) + C_3(n+1) + C_4\left[\frac{(n)(n+1)}{2} + n\right] + C_5\left[n.\frac{(n+1)}{2}\right]$

$= C_1 n + C_1 + C_2 n + C_3 n + C_3 + C_4\left(\frac{n^2+n}{2} + n\right) + C_5\left(\frac{n^2+n}{2}\right)$

$= (C_1+C_2+C_3)n + C_1 + C_3 + C_4\left(\frac{n^2+3n}{2}\right) + C_5\frac{n^2}{2} + \frac{C_5 n}{2}$

$= (C_1+C_2+C_3)n + (C_1+C_3) + \left(\frac{C_4+C_5}{2}\right).n^2 + n\left(\frac{3C_4}{2} + \frac{C_5}{2}\right)$

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

p-7

b)

	Cost	Time
i=1	c_1	1
while i<n	c_2	$\log_2 k$
i = 2*i	c_3	$\log_2 k-1$

$$T(n) = c_1 + c_2 \log_2 k + (\log_2 k - 1)\, c_3$$
$$= (c_2+c_3)\log_2 k + (c_1 - c_3)$$

$$\boxed{T(n) = (c_2+c_3)\log_2 k + (c_1-c_3)}$$

Q5) sol^n

a) $10000n^2 + 5000 = \Theta(n^2)$

As Θ is tight bound & if n increases n^2 will increase rapidly & in such cases constant doesn't matter so

$10000n^2 + 5000 = \Theta(n^2)$ is true

b) $n^3 + n^2 + 100n = \Omega(n^3)$

Max value of above statement is n^3

so

$n^3 + n^2 + 100n = \Omega(n^3)$ is true

c) $\log_n 100 = O(\log n)$

As per O design

$0 \leq f(n) \leq c\,g(n)$ for all $n \geq n_0$

$0 \leq \log n \leq c(\log_n 100)$

$0 \leq \log n \leq c.\log(\log n)$

for any value above statement is true

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

As per O design

$0 \leq f(n) \leq g(n)$

$0 \leq 2^n \leq c.2^{n+1}$

$0 \leq 2^n \leq c.2^n.2^1$

divide eqn by 2^n

$0 \leq 1 \leq c2$

for any value $c>0$ above statement is true

e) $n^3 = O(n^2)$

As per O design

$0 \leq f(n) \leq g(n)$

$0 \leq n^2 \leq c.n^2$

for any value of n above statement will be false

Q5 Soln →

i)

1	3	9	2	8	0	1	5	7	6

↑ i ↑ key j, i+1

$A[i] = 1 > 3$ false

therefore element sorted before key

p-9

ii)

1	3	9	2	8	0	1	5	7	6

i j key i+1

$A[i] = 3 > 9$ (false)

iii)

1	3	9	2	8	0	1	5	7	6

i j key i+1

$A[i] = 9 > 2$ (true)

$A[i+1] = A[i]$ & $i = i-1$

then

1	3	2	9	8	0	1	5	7	6

i j key i+1

$A[i] > key \Rightarrow 3 > 2$ (true)

$\Rightarrow i = i-1$ and $A[i+1] = A[i]$

1	2	3	9	8	0	1	5	7	6

i i+1

$A[i] > key \Rightarrow 1 > 2$ (false)

$A[i+1] = key$

1	2	3	9	8	0	1	5	7	6

• iv)

1	2	3	9	8	0	1	5	7	6
			i	i+1/key					

A[i] > key ⇒ 9 > 8 (true)

→ A[i+1] = A[i] & i = i-1

1	2	3	8	9	0	1	5	7	6
		i	i+1						

A[i] > key ⇒ 3 > 8 (false)

~~then~~ A[i+1] = key

1	2	3	8	9	0	1	5	7	6

• v)

1	2	3	8	9	0	1	5	7	6
				i	i+1 key				

A[i] > key ⇒ i+1

9 > 0 true

A[i+1] = A[i]

i = i-1 — (1)

1	2	3	8	0	9	1	5	7	6

loop (1) continues till

8 > 0, 3 > 0, 2 > 0 & 1 > 0

0	1	2	3	8	9	1	5	7	6

- vi)

0	1	2	3	8	9	1	5	7	6
					q	i+1 key			

A[i] > key ⇒ 9>1 ⇒ True — 2

loop 2 will continues till
8>1, 3>1, 2>1, ⇒ as this true
1>1 false

0	1	1	2	3	8	9	5	7	6

- vii)

0	1	1	2	3	8	9	5	7	6
						q	key i+1		

A[i] > key ⇒ 9>5 ⇒ true — 3

loop 3 will continues till 8>5 ⇒ True
and 3>5 ⇒ false

0	1	1	2	3	5	8	9	7	6

- viii)

0	1	1	2	3	5	8	9	7	6
							q	key i+1	

A[i] > key ⇒ 9>7 true, 8>7 true 5>7 false

0	1	1	2	3	5	7	8	9	6
								q	key i+1

P-12

h)

0	1	1	2	3	5	7	8	9	6

9 (i) — key (i+1)

A[i] > key 9 > 6 true
8 > 6 true
7 > 6 true
5 > 6 False

then

0	1	1	2	3	5	6	7	8	9

Q6 $\xrightarrow{\text{Sol}^n}$

linear Search (Array, Value)
  for i=1 to n
    if (Array[i] == value)
      return i;
  return NIL;

loop invariant :-
For each iteration of for loop $A[i] \neq$ value for every value of i

initialization :-
for i=1 array is single ~~statement~~ element i.e at start of 1^{st} iteration $A[i] \neq$ value

Maintennce :-

let assume at 1st iteration $i=0$, $i=1$ is -1 & we check array[i] == value
if its true. we return index value which i or
we iterate array until [i] become n
As per assumption
if $i=2$ then array [i-1] is not a number which are searching
i.e $i=2$ the array [i] $\neq$ value

Termination :

loop termination if
a) A[i] = value or if
b) $i = n+1$
At $i=n+1$ iteration the element is still not found so loop moment holds true