HOME WORK - 1

Run time

1. a) for $i \leftarrow 1$ to n

    for $j \leftarrow 1$ to n

    for $k \leftarrow j$ to n

    do num $\leftarrow j+1$

return num.

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

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

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

$$= n^3$$

b) for $k \leftarrow 1$ to n

    for $j \leftarrow 1$ to $k * k$

    for $i \leftarrow 1$ to k

    doo num $\leftarrow j+i$

$$\sum_{k=1}^{n} \sum_{j=1}^{n^2} \sum_{i=1}^{n}$$

$$n \sum_{k=1}^{n} \sum_{j=1}^{n^2}$$

$$n \sum_{k=1}^{n} n^2$$

$$n^3 \sum_{k=1}^{n}$$

$$n^3 \cdot n = n^4$$

$$O(n) = n^4$$

② a) $T(n) = cn^4$, where c is a constant.

loop n

for $i \leftarrow 1$ to n : n

for $j \leftarrow 1$ to $n * n$: n^2

for $k \leftarrow 1$ to i : $i = n \Rightarrow n$

do add $\leftarrow i + k$.

return add : $1 = c$

$T(n) = n^2 \cdot n n \cdot c = cn^4$

b, $T(n) = an^3 + bn^2 + cn + d$

loop n

for $i \leftarrow 1$ to n $\quad n$

{ S.O.P("a"); $\quad a$

for $j \leftarrow 1$ to n $\quad n \times n$

{ S.O.P("b"); $\quad b$

for $k \leftarrow 1$ to n $\quad n \times n \times n$

{ S.O.P("c"); $\quad c$

}

}

} S.O.P("d"); $\quad d.$

$\therefore T(n) = an^3 + bn^2 + cn + d.$

c, $T(n) = c \log n$

Recursive Binary Search (A, v, low, high)

if low > high

return Null

mid = ((low + high)/2)

if $V == A[mid]$

return mid.

else if $V > A[Mid]$

return Recursive - Binary Search $(A, V, mid-1, low)$.

else return Recursive - Binary Search $(A, V, low, high)$

return $V \rightarrow$ Constant in c

3, Use the master theorem to Solve the following

Master's Theorem :-

$T(n) = a\ T(n/b) + \theta(n^k \log^p n)$

$a \geq 1$, $b > 1$, $k \leq 0$ & p is real number

1. if $a > b^k$, then $T(n) = \theta(n^{\log_b a})$

2. If $a = b^k$

a) if $P > -1$, then $T(n) = \theta(n^{\log_b a} \log^{P+1} n)$

b, if $P = -1$, then $T(n) = \theta(n^{\log_b a} \log \log n)$

c, if $P < -1$, then $T(n) = \theta(n^{\log_b a})$

3. if $a < b^k$

a) if $p \geq 0$, then $T(n) = \Theta(n^k \log^p n)$

b) if $p < 0$, then $T(n) = \Theta(n^k)$

a) $T(n) = 16T(n/4) + n!$

Here $a = 16$, $b = 4$, $k = 1$, $p = 0$

$16 > 4^{(1)}$

It comes under case 1

$T(n) = \Theta(n^{\log_b a}) = \Theta(n^{\log_4 16})$

$= \Theta(n^{\log_4 4^2}) = \Theta(n^2)$

b) $T(n) = 3T(n/3) + n/2$

Here $a = 3$, $b = 3$, $k = 1$, $p = 0$

$3 = 3^1$

It comes under case 2(a)

$T(n) = \Theta(n^{\log_b a} \log^{p+1} n)$

$= \Theta(n^{\log_3 3} \log^1 n) = \Theta(n \log n)$

Scanned by CamScanner

4) $T(n) = 4T(n/2) + n^2 \log n$

Here $a=4$, $b=2$, $k=2$, $p=1$

$4=4$

It comes under case 2(a)

$T(n) = \theta(n^{\log_b a} \log^{p+1} n)$

$= \theta(n^{\log_2 4} \log^2 n)$

$= \theta(n^2 \log^2 n)$.

5) By using Insertion Sort, Sort $A = \{4, 10, 8, 9, 12, 15, 1\}$

Step 1:

4	10	8	9	12	15	1

Step 2:

4	10	8	9	12	15	1

Step 3:

Step 4:

4	8	10	9	12	15	1

Step 5:

4	8	9	10	12	15	1

Step 6:

4	8	9	10	12	15	1

Step 7:

4	8	9	10	12	15	1

Sorted list

1	4	8	9	10	12	15

6, Binary Search Pseudocode.

Binary Search (S, K, low, high)

a). If low > high

return no such key

* Best case : O(1)

* Worst case : O(nlogn)

Else

mid ← (low + high)/2

If K = key (mid) then

return key (mid)

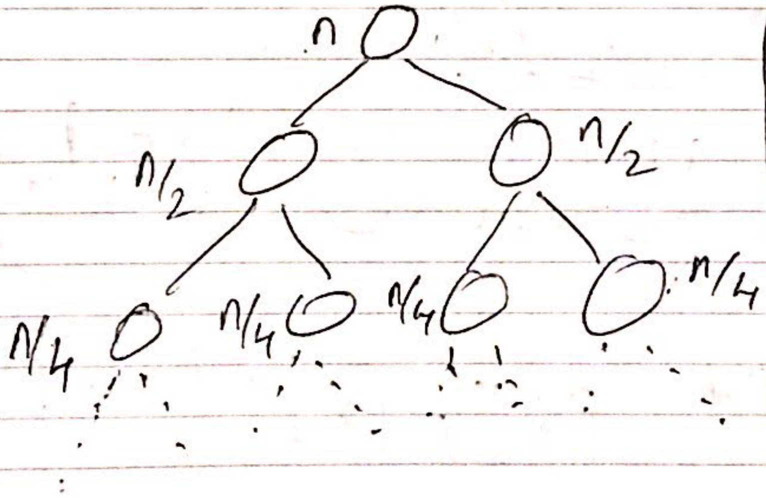
else if K < key (mid) then

return Binary Search (S, K, low, mid-1)

else.

return Binary Search (S, K, mid+1, high)

b) Running time of Binary Search



On one comparision Binary search tree reduces or eliminates half of the elements.

Comparision — Search Range

Comparision	Search Range
0	n
1	$n/2$
- - -	- - - -
2^i	$n/2^i$
$\log_2 n$	1

$\therefore$ Binary Search runs in $O(\log n)$ time.

7) Rank the following functions by order of growth.

n, $\log\log n$, $n!$, n^2, $\cancel{A}$ 1, n^n, $2^{\log n}$, n^3, n^k, 2^n

Ans: The Ranking is:

$1 < \log\log n < n < 2^{\log n} < n! < n^2 < n^3 < n^k < 2^n < n^n$

4) Recurrence $T(n) = 3T(n/2) + cn^2,\ n \geq 1$

a) Cost at levels 0, 1, 2, 3

cn^2

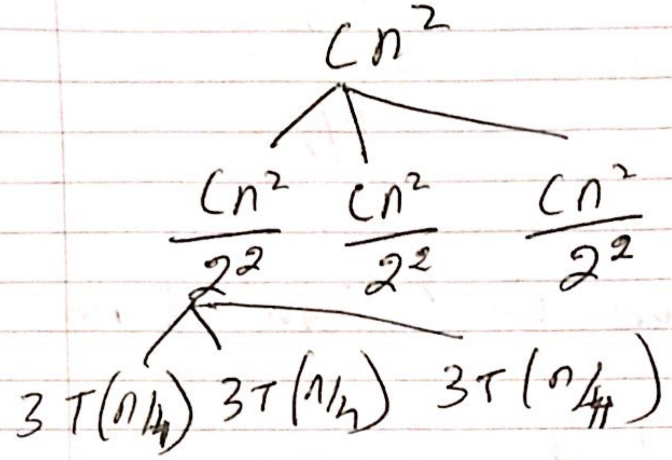
$T(n/2)\quad T(n/2)\quad T(n/2) \longrightarrow$ 1 level

$$T(n/2) = 3T(n/4) + c(n/2)^2$$

$$T(n) = 3\left[3T(n/4) + c(n/2)^2\right] + cn^2$$

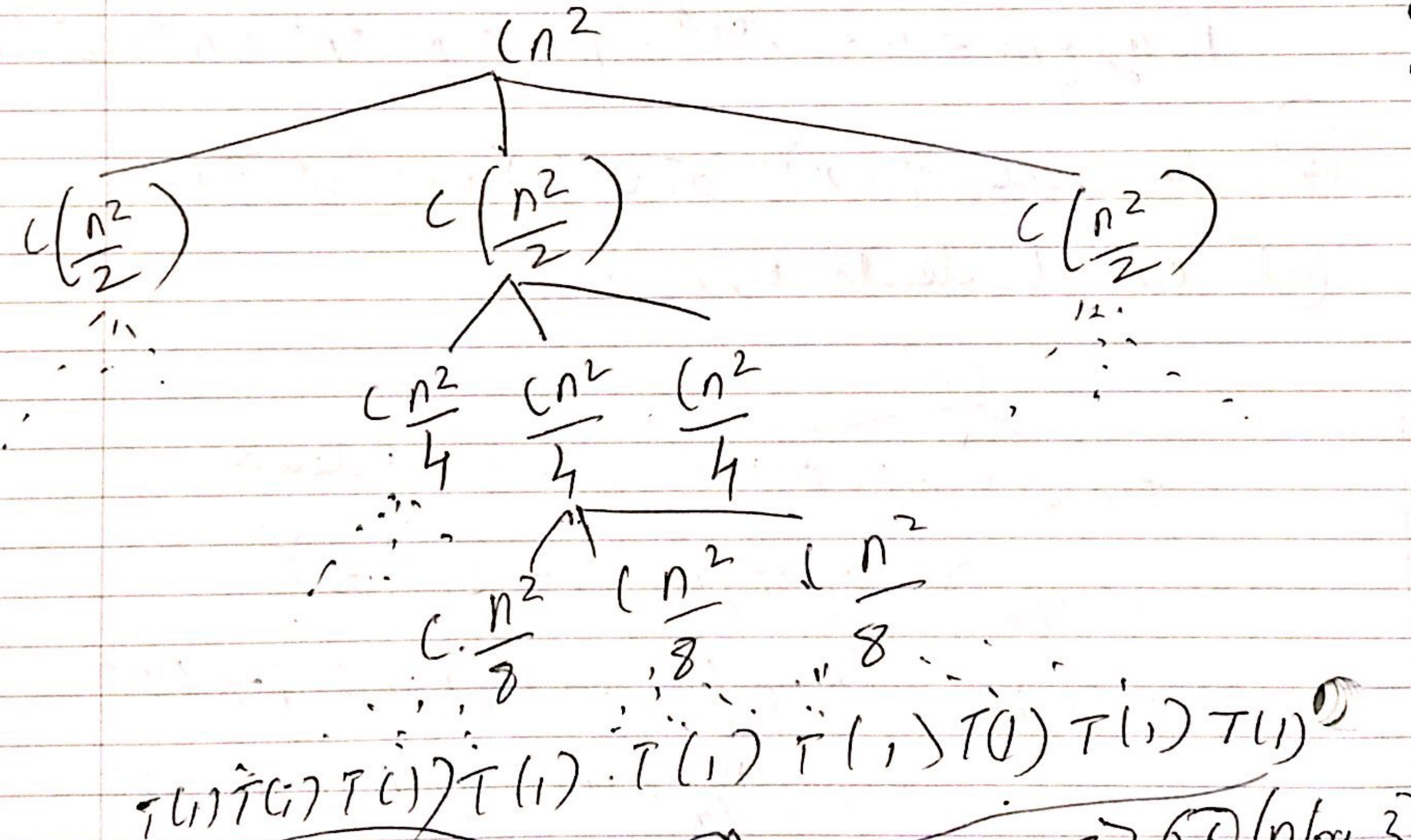
$$= 9T(n/4) + 3c(n/2)^2 + cn^2$$

$$= 9T(n/4) + \frac{5cn^2}{2}$$



$$T(n/4) = 3T(n/8) + c(n/4)^2$$

$$T(n/8) = 3T(n/16) + c(n/8)^2$$



$$n\log_2 3 \Rightarrow \Theta(n\log_2 3)$$

b) Height of Tree

SubProblem Size for level 0 → $n/2$
SubProblem Size for level 1 → $n/2$
Eventually the SubProblem Size for node at depth i is $n/2^i$

$n/2^i = 1$

$2^i = n$
$i = \log_2 n$

height of Tree $= \log_2 n$
$= \log_2 8^4 = 12$

$2^x = 8^4$
$2^x = (2^3)^4$
$2^x = 2^{12}$
$x = 12.$

(c) $T(n)$ as sum of each cost in level.

$$T(n) = cn^2 + \frac{3}{4}cn^2 + \left(\frac{3}{4}\right)^2 cn^2 + \left(\frac{3}{4}\right)^2 cn^2 \ldots + \left(\frac{3}{4}\right)^{\log_2 n-1} + x \cdot n^{\log_2 3}$$

[No of levels → $0, 1, 2, 3, \ldots (\log_2 n - 1)$]

$$= \sum_{i=0}^{\log_2 n-1} \left(\frac{3}{4}\right)^i cn^2 + x \cdot n^{\log_2 3}$$

Scanned by CamScanner

d) $T(n) = \sum_{i=0}^{\log_2 n-1} \left(\frac{3}{4}\right)^i cn^2 + \Theta(n^{\log_2 3})$

$< \sum_{i=0}^{\infty} \left(\frac{3}{4}\right)^i cn^2 + \Theta(n^{\log_2 3})$

$= \frac{1}{1-(3/4)} cn^2 + \Theta(n^{\log_2 3})$

$= \frac{4}{3} cn^2 + \Theta(n^{\log_2 3})$

$= O(n^2)$

e $T(n) = 3T(n/2) + cn^2 \quad n > 1$

Substitution Method

$T(n) = 3T(n/2) + cn^2$

$T(n/2) = 3T(n/4) + c(n/2)^2$

$T(n) = 3\left[3T(n/4) + c(n/2)^2\right] + cn^2$

$= 9(n/4) + 3\frac{n^2}{4} + cn^2$

$T(n/4) = 3T(n/8) + c(n/4)^2$

$$T(n) = 9\left(3T(n/3) + c(n/4)^2\right) + \frac{3}{4}\frac{cn^2}{4} + cn^2$$

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

$$= n \sum_{i=0} (3/4)^i cn^2$$

$$\leq \sum_{i=0}^{\infty} (3/4)^i cn^2$$

$$= \frac{1}{1-3/4} n = 4n.$$