142103002

Tutorial 2

Q.1. $T(n) = 8T(n/2) + n^2$

Solve by using Recursion tree method.

$T(n)$ — executing time

$n/2 \quad \frac{n}{2} \quad n/2 \quad n/2 \quad n/2 \quad n/2 \quad n/2 \quad n/2$ — $8(n/2)^2 = 2n^2$

$n/4$ — $8^2(n/4)^2 = 4n^2$

$n/8$

$n/2^k \ldots n/2^k \ldots$ — $2^k n^2$

Height of the tree:-

$n/2^k = 1 \; ; \; k = \log_2 n$

No. of leaves $= 2^k n^2$

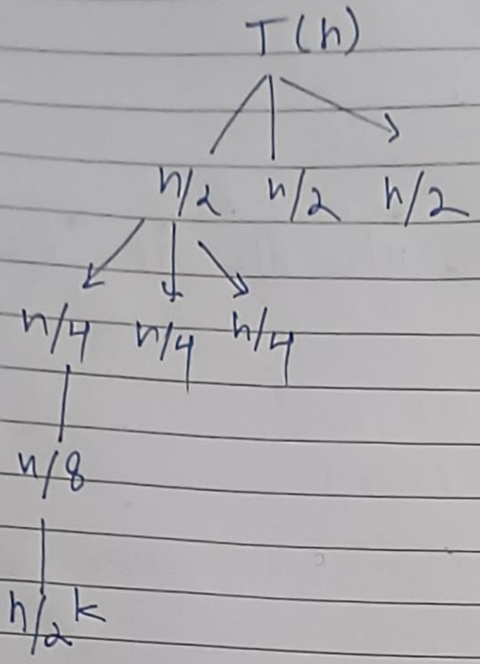
$= 2^{\log n} \cdot n^2$

$= n \cdot n^2 = n^3$

$\therefore T(n) = O(n^3)$

142103002

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



Height of the tree =

$\frac{n}{2^k} = 1$; $k = \log_2 n$

Number of leave Nodes -

$\left(\frac{3}{2}\right)^k \cdot n = \left(\frac{3}{2}\right)^{\log_2 n} \cdot n = \frac{3^{\log_2 n}}{2^{\log_2 n}} \cdot n$

$= 3^{\log_2 N}$ $(\because a^{\log_a b} = b)$

$= n^{\log_2 3}$ $(\because c^{\log_a b} = b^{\log_a c})$

total time cost $= (\underbrace{n + 3/2\, n \ldots n^{\log_2 3}}_{\log_2 n \text{ times}})$

$= \frac{n\left(\left(3/2\right)^{\log_2 n} - 1\right)}{\left(3/2 - 1\right)} = 2 \cdot \left(3^{\log_2 N} - n\right)$

$= 2n^{\log_2 3} - 2n = 2n^{1.53} - 2n = \underline{O(n^2)} \therefore T(n)$

14203002

→ $T(n) = 2T(n/2) + n^2$

solve by using master's theorem.

$f(n) = n^2$

$a = 2 \quad b = 2$

$f(n) = O(n^k \log^p n)$

$n^2 = O(n^k \log^p n)$

$k = 2, \; p = 0$

→ $T(n) = 2(n-1) + 1$

solve by using substitution method.

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

$T(n-2) = 2T(n-3) + 1$

so,

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

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

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

$= 2^2[2T(n-3) + 1] + 2 + 1$

$= 2^3 T(n-3) + 2^2 + 2 + 1$

$T(n) = 2^k T(n-k) + (1 + 2 + 2^2 + \cdots + 2^{k-1})$

g.p. with ar = 1(2) a = 1 r = 2

n; no of elements

sum of g.p. $= \frac{a(r^n - 1)}{(r-1)} =$

$T(n) = 2^k T(n-k) + \frac{1(2^k - 1)}{(2-1)}$

$T(n) = 2^k T(n-k) + 2^k - 1$, let $n = k$

$T(n) = 2^n + (10) + 2^n - 1 = 2(2^n) - 1) = O(2^n)$.

142103002

Q.2.

→ If $f(n) = n^{\sqrt{2}\log n}$, $g(n) = 2^{\sqrt{n}}$ then $f(n) > g(n)$

$f(n) > g(n)$

$2^{\log n} : n^{\log n}$

Taking log on both sides

$(\log n).(\log 2) : (\log n)(\log n)$

$(\log n) : (\log n)^2$

for large values of n.

$\log n < (\log n)^2$

$f(n) < g(n)$

so,

$f(n) > g(n)$ is false.

- $n^2 \cdot 2^{3\log n} = \Theta(n^5)$

LHS $= n^2 . 2^{\log(n^3)}$ $(\because a^{\log_a(b)} = b)$

$= n^2 . n^3 = n^5$

$\Theta(n5)$

LHS = RHS

True.

142103002

- $4^n / 2^n = \theta(2^n)$

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

$\frac{2^{2n}}{2^n} = 2^n \quad \therefore \theta(2^n).$

- If $f(n) = n^{\sqrt{2}.\log n}$, $g(n) = 2^{\sqrt{n}}$ then $f(n) > g(n)$
$f(n) > g(n)$

$n^{\sqrt{2}.\log n} : 2^{\sqrt{n}}$

Taking log on both sides;

$\sqrt{2}.\log n . \log n : \sqrt{n}.\log 2$

$\sqrt{2}.(\log n)^2 : \sqrt{n}$

$2 \log n : n$

For large values of n

$2 \log(n)^4 < n$

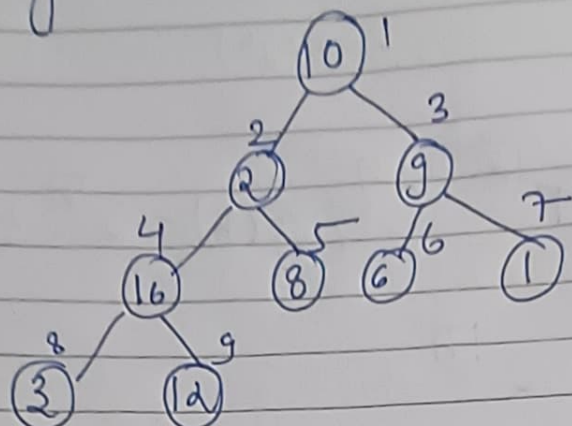
so,

$F(n) < g(n)$

142103002

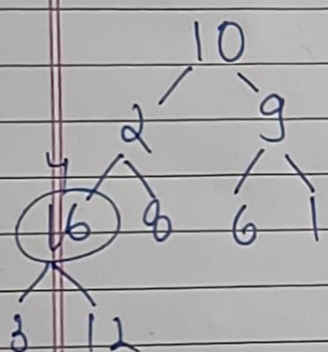
Q3 A[1...9] = {10, 2, 9, 16, 8, 6, 1, 3, 12}
maxheap

Array Before =>



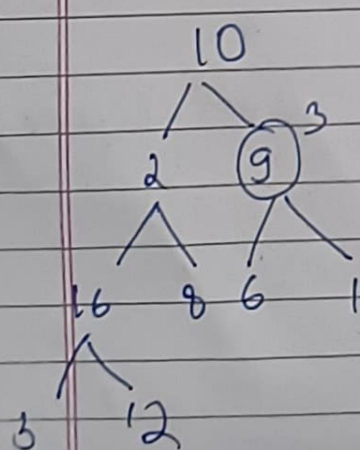
last non-leaf node = [n/2] = 4
we will start hepifying from i = 4th element

For = i = 4



=> No hepify function req.

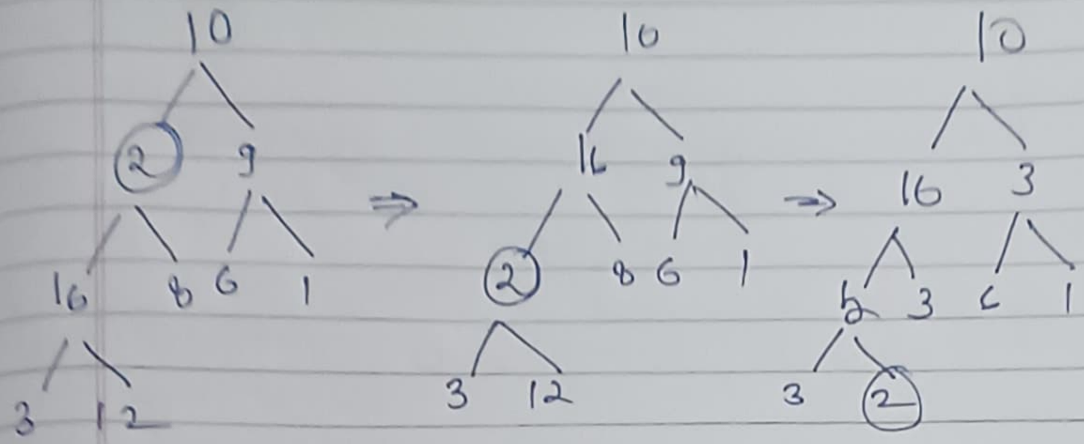
For i = 3



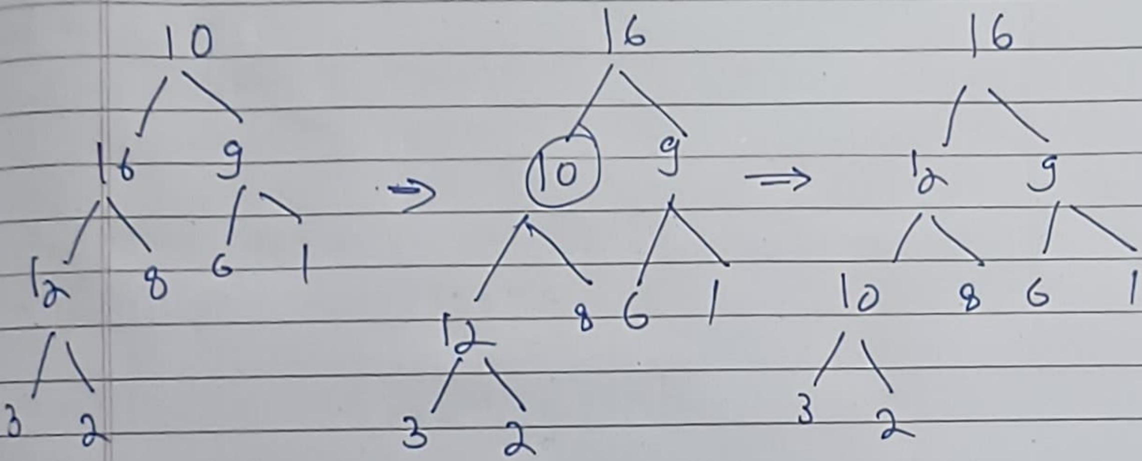
=> No heapify req.

142103002

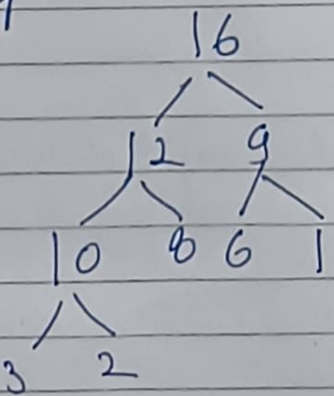
for i = 2



for i = 1



final heap.



A = {16, 12, 9, 10, 8, 6, 1, 3, 2}