1

a) $T(n) = 2.T(n/2) + n^3$ using Substitution method

Guess = $T(n) = O(n^3)$ since we have $\boxed{+n^3}$

1-induction base: $T(1) \leq c1^3 \leq c$

we can always pick c big enough to satisfy this

Assume = $T(k) \leq ck^3$ for all $k < n$ ← 2. Inductive step.

induction hypothesis → $T(n) \leq cn^3$.

we can say $\frac{k}{2} < \frac{n}{2}$ since $k<n$ is the assumption

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

$\Rightarrow 2.T(k/2) + n^3 \rightarrow \leq 2.c\left(\frac{k}{2}\right)^3 + n^3$ since n is bigger than k we can do next step.

$\leq 2.c.\left(\frac{n}{2}\right)^3 + n^3$

$\leq \frac{c}{4}.n^3 + n^3 \rightarrow \underbrace{\frac{c+4}{4}}_{\text{constant } c_2} . n^3$ which looks like $T(n) \leq c_2 n^3$

so IH is true $T(n) = \theta(n^3)$

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

Guess = $T(n) = O(n^{\log_2 7})$ by master method.

1-Induction base: $T(1) \leq c.1^{\log_2 7} \leq c$

2 Induction step:

- Assumption = $T(k) \leq ck^{\log_2 7}$ for all $k<n$
- Ind. Hypothesis = $T(n) \leq cn^{\log_2 7}$

$T(n) = 7T(\frac{n}{2}) + n^2$

$\leq 7.c.\left(\frac{n}{2}\right)^{\log_2 7} + n^2 \rightarrow \leq 7.c.\frac{n^{\log_2 7}}{2^{\log_2 7}} + n^2 \rightarrow c.n^{\log_2 7} + n^2$ → is not proven

So, Assume = $T(k) \leq c_1 k^{\log_2 7} - c_2 k$ for all $k<n$

Ind. Hypothesis is the same.

$T(n) = 7.\left\{c_1.\left(\frac{n}{2}\right)^{\log_2 7} - c_2.\left(\frac{n}{2}\right)\right\} + n^2$

$\leq c_1.n^{\log_2 7} - 7.c_2.\frac{n}{2} + n^2$

$\leq c_1.n^{\log_2 7} - \underbrace{\left\{\left(\frac{7.c_2}{2} - n\right).n\right\}}_{\text{Residual}}$

so $T(n) = \theta(n^{\log_2 7})$

Scanned with CamScanner

(2)

c) $T(n) = 2T(n/4) + \sqrt{n}$

Master method $a = 2, b = 4$ $\log_4 2 = \frac{1}{2}$ so $f(n) = n^{\frac{1}{2}}$

Case 2: if $f(n) = \Theta(n^{\log_b a}) \rightarrow T(n) = \Theta(n^{\log_b a} . \log n)$ so,

Guess: $T(n) = \Theta(n^{\frac{1}{2}} . \log n)$

1. IB: $T(1) \leq c . 1^{\frac{1}{2}} . \log 1 \leq c$ holds. for all constants.

2. IS:

- Assumption: $T(k) \leq c . k^{\frac{1}{2}} . \log k$ for all $k < n$
- IH: $T(n) \leq c . n^{\frac{1}{2}} . \log n$

$$T(n) = 2 . c . \left(\frac{n}{4}\right)^{\frac{1}{2}} . \log \left(\frac{n}{4}\right) + \sqrt{n}$$

$$\leq c . \sqrt{n} . (\log n - 2) + \sqrt{n} \rightarrow \underbrace{c . n^{\frac{1}{2}} . \log n}_{\text{what we want}} \underbrace{- 2 . c . n^{\frac{1}{2}} + n^{\frac{1}{2}}}_{\text{residuals}}$$

so, $T(n) = \Theta(n^{\frac{1}{2}} . \log n)$

d) $T(n) = T(n-1) + n$ (iteration method.)

$= T(n \quad + i n -$

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

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

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

$T(n) = T(n-i) \; i . n + \sum_{k=0}^{i-1} -k$

$T(n) = i . n + \sum_{k=0}^{i-1} -k$

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

$n = 2$, $i = n$

Note that $i = n \rightarrow T(0)$

hence the recursion will bottom out when $i = n$

so, $T(n) = T(0) + n^2 + \underbrace{\sum_{k=0}^{n-1} -k}_{\text{residual}}$

so $Tn = \Theta(n^2)$

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

Guess = $T(n) = O(2^n)$

Assume = $T(k) \leq c2^k$ for all $k < n$

$T(n) \leq c2^n$

$k-1 \leq n-1$

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

$T(n) = 2 \cdot c \cdot 2^{n-1} + 2n-2$

$\leq 2 \cdot c \cdot \frac{2^n}{2} + 2(n-1)$ not done

So, Assume $T(k) \leq c_1 2^k - c_2 k$ for all $k < n$

$T(n) \leq c_1 2^n - c_2 n$

$T(n) = 2 \cdot (c_1 2^{n-1} - c_2(n-1)) + 2(n-1)$

$T(n) = c_1 \cdot 2^n - 2c_2(n-1) + 2(n-1)$

$T(n) = c_1 \cdot 2^n - \underbrace{(2-2c_2)(n-1)}_{\text{residual when } c_2 < 1}$

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