Mid Term-1 Solution

Question 1:

a) (0.7 points)

Algorithms	Worst Case	Best Case
Insertion Sort	$\boldsymbol{O(n^2)}$	$\boldsymbol{O(n)}$
Merge Sort	$\boldsymbol{O(nlogn)}$	$\boldsymbol{O(nlogn)}$
Heap Sort	$\boldsymbol{O(nlogn)}$	$\boldsymbol{O(nlogn)}$
Quick Sort	$\boldsymbol{O(n^2)}$	$\boldsymbol{O(nlogn)}$
Counting Sort	$\boldsymbol{O(n+k)}$	$\boldsymbol{O(n+k)}$
Radix Sort	$\boldsymbol{O(nk)}$ **or O(d(n+k))**	$\boldsymbol{O(nk)}$ **or O(d(n+k))**
Bubble Sort	$\boldsymbol{O(n^2)}$	$\boldsymbol{O(n)}$

b) Show $3n^2 + 4n - 3 = O(n^2)$. **(0.8 points)**

We need to find c and n_0 such that:
$3n^2 + 4n - 3 <= cn^2$ for all $n >= n_0$
Divide both sides by n^2, getting:
$3 + 4/n - 3/n^2 <= c$ for all $n >= n_0$
If we choose n_0 equal to 1, then we need a value of c such that:
$3 + 4 - 3 <= c$
We can set c equal to 4. Now we have:
$3n^2 + 4n - 2 <= 4n^2$ for all $n >= 1$

Or
$3n^2 + 4\,n^2 - 3\,n^2 <= cn^2$; either c = 4 or 7 is true for n = 1

Question 2:

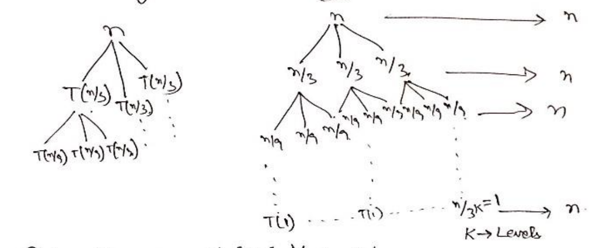
a. (2.5 points) **b. (1 points)**

Q No: 2

(a) The recurrence is

$$\left.\begin{aligned} T(n) &= 3T(n/3) + n \\ T(1) &= 1 \end{aligned}\right\} \quad 0.5 \text{ points}$$

Solution using recurrence tree $\Rightarrow$ 2 points



Subproblem size at level K is $n/3^K$.

Subproblem size hits 1 when $n/3^K = 1 \Rightarrow K = \log_3 n$.

Cost at each level is n or $O(n)$.

Number of nodes at each level $3^K \Rightarrow 3^{\log_3 n} = n^{\log_3 3} = n$.

Total Cost

$$T(n) = \sum_{K=0}^{\log_3 n} n = \log_3 n \times n = n\log n$$

So, Cost is $O(n\log n)$.

OR

As cost at each level is n & we have to K-level

So Total Cost is $n \times K \Rightarrow O(nK) = O(n\log n)$

Q No: 2

(b) $T(n) = \sqrt{n}\, T(\sqrt{n}) + n$ Solution $\Rightarrow O(n\log\log n)$

Induction Goal

$T(n) \le dn^2$ for some d & $n \ge n_0$

$T(\sqrt{n}) \le d\sqrt{n} \log\log\sqrt{n}$.

Proof:

$$\begin{aligned} T(n) &= \sqrt{n}\, T(\sqrt{n}) + n \\ &\le \sqrt{n}\cdot d\sqrt{n} \log\log\sqrt{n} + n \\ &= dn \log\log (n)^{1/2} + n \\ &= dn \log(\tfrac{1}{2}\log n) + n \\ &= dn\log\tfrac{1}{2} + dn\log\log n + n \\ &= dn\log\log n + dn(\log 1 - \log 2) + n \\ &= dn\log\log n + dn \times 0 - dn\log 2 + n \\ &= dn\log\log n - dn + n \\ &\le dn\log\log n \quad \text{(This is true when } -dn + n \le 0) \end{aligned}$$

So $-dn + n \le 0$

$-d + 1 \le 0$

$\boxed{d \ge 1}$

Now the value of n; n should be $n \ge 4$ to hold the condition ~~as on the value of 1 or 2.~~

$\boxed{T(n) \le dn\log\log n}$ Hence proved

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

// Divide both sides by n

T(n)/n = 3T(n/3)/n + 1

T(n)/n = T(n/3)/(n/3) + 1

T(n)/n = T(n/9)/(n/9) + 1 + 1

T(n)/n = T(n/27)/(n/27) + 1 + 1 + 1

T(n)/n = T(n/3^3)/(n/3^3) + 3

In general

T(n)/n = T(n/3^k)/(n/3^k) + k

Let k = log$_3$n

T(n)/n = T(1) + log$_3$n => T(n) = n + nlog$_3$n => T(n) =O(nlogn)

Question 3: (3 points)

Compute the time complexity of the provided recursive relations by using master theorem and write it in the most appropriate asymptotic notation.

The generic form of the equation is as $T(n) = aT\left(\frac{n}{b}\right) + f(n)\ where\ f(n) = \theta(n^c)$

a) $\boldsymbol{T(n) = 4T(n/4) + 4n^2}$

$a = 4, b = 4, c = 2$

a < b^c:

$c > \log_b a$

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

$\boldsymbol{T(n) = \theta(n^c) = \theta(n^2)}$

b) $\boldsymbol{T(n) = T(n/2) + 8}$

$a = 1, b = 2, c = 0, i.e. a = b^c, f(n) = \theta(n^2 \log \quad n)$

$c = \log_b a$

$T(n) = \theta(n^c \log \quad n)$

$\boldsymbol{T(n) = \theta(\log n)}$

c) $\boldsymbol{T(n) = 16T(n/2) + 16n}$

$a = 16, b = 2, c = 1\ i.e. a > b\hat{}c$

$c < \log_b a$

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

$\boldsymbol{T(n) = \theta(n^{\log_2 16}) = \theta(n^4)}$

d) $\boldsymbol{T(n) = 2T\left(\frac{n}{2}\right) + 2n \log n}$

Special Case:

If $f(n) \in \Theta(n^{\log_b a} \log^k n)$ *for some* $k \geq 0$ *then*

$$T(n) \in \Theta(n^{\log_b a} \log^{k+1} n)$$

$a = 2, b = 2, c = 1, f(n) = \theta(n^c \log^k n)$

$c = \log_b a$

$T(n) = \theta(n^c \log^{k+1} n)$

$\boldsymbol{T(n) = \theta(n \log^2 n)}$

Question 4: (2 points)

```
carry = 0;
for(i=N-1 to 0) do
r[i+1] = (a[i] + b[i] + carry) % 10;
carry = (a[i] + b[i] + carry)/10;
end for
r[0] = carry;
```