Problem 2

PAGE No.
DATE / /

1) for inserting a new element in Amortized dictionary,
- we need to add new element in temporary array of size 1 and then iterate through all arrays from $i=0$ to $i=k$ (where, $k = \lceil \log(n+1) \rceil$ arrays.)
- we need to check if A_i is empty, if yes then copy temporary array in A_i & stop. Elese, merge temporary array with A_i and increment array till $i=k$.

Now, for merging two sorted lists of same size, it takes twice comparisons to the size of array.
ie. for two arrays of size l,
$2l$ comparisons are required. ..(1)

$\therefore$ Merge cost for A_i will be 2×2^i
... considering point (1) in worst case.

Factualising above formula, for $i = 0, 1, 2 \ldots k \rightarrow$
$A_i = \frac{n}{2^i}$ times cost of 2^i

ie $A_0 = \frac{n}{2}$ times with cost of 2

$A_1 = \frac{n}{4}$ times with cost of 4 .. & so on.
... $A_{k-1} = \frac{n}{2^{k-1}}$

$\therefore$ Total cost $= O(n \log n)$ for n insertions.
Therefore, amortized cost for insertion correspondence to $O(\log n)$.

PAGE No.
DATE / /

Problem 2

2) For every 'n' number of elements, we have 'k' number of arrays ~~where~~ ~~k~~ with each array size incresing by powers of 2.
K corresponds to $\log(n+1)$

To perform search operation, we need to check for the key in each array one by one

Since we have 'k' number of arrays, in worst case all 'k' arrays will be searched.

To search in each array particularly, since the array is sorted, it will take $O(\log t)$ time to search ~~for~~ in array of t elements using binary search.
$\therefore$ size of longest array will always be less than n
ie $len(A_{k-1}) \leq 1$
$\therefore$ Each search $\approx O(\log n)$
$\therefore$ For overall search time complexity is $k \cdot \log n \in O(\log^2 n)$.

DATE / /

Problem 3

1) $T(n) = 8 * T(n/3) + 2^n$

Comparing given equation with standard eqn.

$T(n) = aT(n/b) + f(n)$, $f(n) = 2^n$, $a = 8$, $b = 3$

To prove - $f(n) = n^{\log_b a}$ ... 1)

substituting values of a & b in 1)

$= n^{\log_3 8}$

$= n^{\log_3 5 + 3}$ ie $n^{\log_3 3 + 5}$

comparing with case 3 of Master's theorem,

$\epsilon = 5$ $\therefore$ $f(n) = n^d$

Now, calculating for $af(n/b)$

to prove - $8 \times 2^{n/3} \leq cf(n)$ ... 2)

where $c < 1$

lets consider $c = 0.5$

$\therefore$ Eqn. 2) can be written as -

$$2^3 \times 2^{n/3} \leq \frac{1}{2} \cdot 2^n$$

$$= 2^{9 + n/3} \leq 2^{n-1}$$

$$\therefore \frac{9+n}{3} \leq n-1$$

substituting value of $n = 6$

$\therefore$ LHS $= \frac{15}{3} \approx 5$ and RHS $= 5$

also, for $n = 7$

LHS $= \frac{16}{3} \approx 5$ and RHS $= 6$

$\therefore$ LHS = RHS

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

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

Problem 3

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

→ comparing given equation with standard equation- $T(n) = aT(n/b) + f(n)$,

$f(n) = \frac{n}{2} = \frac{1}{2}n$, $a = 3$, $b = 3$

$f(n) \approx n^{\log_b a}$

$= n^{\log_3 3}$

$\therefore \cancel{f(n)} = n^1$ --- 1)

The above equation is equivalent to Master's theorem equation -

~~$f(n) = \theta\{n\log$~~

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

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

from 1), $T(n) = \theta(n\log n)$

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

→ comparing given equation with standard equation- $T(n) = a\{T(n/b) + f(n)$,

$f(n) = \sqrt{n}$, $a = 2$, $b = 4$

$f(n) = \sqrt{n}$ ... ①

from Master's theorem,

$f(n) = n^{\log_b a}$ --- ②

Substituting valese of eq^n ① in ②,

~~$f(n) = \sqrt{n}^{\log_4 2}$~~

~~$= \sqrt{n} \times \frac{1}{2}$~~

$\cancel{f(n)} = n^{\log_4 2}$ ... (substitute valeses of a & b)

$= n^{1/2}$ --- ($\log_4 2 = \frac{1}{2}$)

$\cancel{f(n)} = \sqrt{n}$ .... ③

Equation ③ is equivalent to Master's theorem eq^n

PAGE No.
DATE / /

$f(n) = \Theta n^{\log_b a}$ then,

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

$\therefore$ for $f(n) = \sqrt{n}$,

$T(n) = \Theta(\sqrt{n} \log n)$

4). $T(n) = 4 \times T(n/2) + 3 \times n$

$\rightarrow$ comparing given eq^n with standard equation - $T(n) = aT(n/b) + f(n)$,

$f(n) = 3n$, $a = 4$, $b = 2$

Using Master's Theorem,

~~f(f(n))~~ $= n^{\log_b a}$

$= n^{\log_2 4}$ .. substitute values of a&b.

$\therefore$ ~~f(n)~~ $= n^2$ ... $(\log_2 4 = 2)$ ... ⓐ

Above equation is equivalent to Master's Theorem equation -

If $f(n) = \Theta(n^{\log_b a - \epsilon})$ then,

$T(n) = \Theta(n^{\log_b a})$ ... 1)

since $\epsilon = 1$ satisfies above equation 1),

$\therefore$ from case 1) of Master's Theorem,

$T(n) = \Theta(n^2)$ ... from ⓐ and 1).

PAGE No.
DATE / /

Problem 4

4) Let's consider c_i = cost of i^{th} operation

$\therefore$ $c_i = i$, where i is perfect square.

$= 0$, otherwise.

Analysing cost for each operation -

Operation	Cost
1	1
2	0
3	0
4	4
5	0
⋮	⋮

, and so on.

$\therefore$ for n operations, the cost can be calculated as-

$$\sum_{i=1}^{n} c_i \leq \sum_{i=1}^{n} \sqrt{i}^{\,2} \leq n$$

$$\therefore \cancel{\sqrt{n}^2} \neq \cancel{\frac{n(n+1)}{2}} \cancel{\leq n} \qquad \therefore \sqrt{n}^2 = \frac{(\sqrt{n})(\sqrt{n}+1)(2\sqrt{n}+1)}{6}$$

$$= n\sqrt{n} \cancel{\leq n}$$

$\therefore$ $c_i = n\sqrt{n}$

$\therefore$ Average cost using aggregate analysis is-

$$\therefore \cancel{\frac{T(n)}{\sqrt{n}}} = \cancel{T(\sqrt{n})}$$

$$\therefore \frac{T(n\sqrt{n})}{\cancel{\sqrt{}}\,n} = \underline{\underline{T(\sqrt{n})}}$$