CS-DSA 4413

HW # 3 Solutions

Q1

Sol:- Directly from slides. (Module 3, Slide 13 to 17)

In Slide 16, $\sum_{i=1}^{k} i = \frac{k(k+1)}{2}$ — (2)

Applying (2) to (1),

$$T(n) = \frac{n(n-1)}{2} \quad \left[\because T(n) = \sum_{i=1}^{n-1} i\right]$$

$$= \frac{n^2 - n}{2}$$

Polynomial of degree 2.

Q2

Sol:- We know that given functions are asymptotically non negative.

By applying Θ definition,

$$0 \leq c_1 (f_1(x) + f_2(x) + f_3(x) + f_4(x) + f_5(x))$$

$$\leq \max(f_1(x), f_2(x), f_3(x), f_4(x), f_5(x))$$

$$\leq c_2 (f_1(x) + f_2(x) + f_3(x) + f_4(x) + f_5(x))$$

for $c_1 = \frac{1}{5}$, $c_2 = 1$, The above inequality holds good. Hence proved.

Q3

Sol:- We have an array 'A' with 'k' integers.

(i) Use any sorting techinque with $\Theta(n \log n)$.
After this step, we will get 'k' integers in sorted order.

(ii) Then for each integer 'x' in array 'A' use binary search to check if integer 'z-x' exists in array 'A'. Binary search takes $O(\log k)$ and is executed 'K' times. So total $O(n \log n)$.

Q4 (a) $T(n) = T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + 1$, $T(1) = 1$, $n = 2^k - 1$.

Sol

Take any n, where $n = 2^k - 1$,
for example, $15 = 2^4 - 1$, $T(15) = T\left(\left\lfloor \frac{15}{2} \right\rfloor\right) + 1$

$T(7) = T\left(\left\lfloor \frac{7}{2} \right\rfloor\right) + 1$

$T(3) = T\left(\left\lfloor \frac{3}{2} \right\rfloor\right) + 1$

$T(3) = T(1) + 1$

~~$T(n) = T\left(\left\lfloor \frac{2^k - 1}{2} \right\rfloor\right)$~~

$$T(n) = T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + 1 \quad -(1)$$

$$T\left(\frac{n}{2}\right) = T\left(\left\lfloor \frac{n}{2^2} \right\rfloor\right) + 1 \quad -(2)$$

$$T(n) = T\left(\left\lfloor \frac{n}{2^2} \right\rfloor\right) + 1 + 1 \quad -(3)$$ [Obtained by substituting (2) in (1)].

After (k-1) iterations.

$$T(n) = T\left(\left\lfloor \frac{n}{2^{k-1}} \right\rfloor\right) + (k-1).$$

(b) $T(n) = T(n^{\frac{1}{2}}) + 1, \quad n = 2^{2^k}, \quad T(2) = 1$

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

$$T(n^{\frac{1}{2}}) = T(n^{\frac{1}{4}}) + 1 \quad -2$$

Eq(2) is obtained by substituting n with $n^{\frac{1}{2}}$ in eq(1).

$$T(n) = \underbrace{T(n^{\frac{1}{4}}) + 1}_{T(n^{\frac{1}{2}})} + 1 \quad -1$$

By Sub 2 in 1.

$$T(n) = T(n^{\frac{1}{\cancel{2}}}) + \cancel{8}$$ → Generalized equation.

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

$$n^{\frac{1}{2^k}} = 2 \Rightarrow k = \log_2(\log_2 n)$$

$n^{\frac{1}{2^r}} = 2,$

$\log n = 2^r.$

$2^k = 2^r$

$\Rightarrow k = r.$

$$T(n) = T(2) + \log_2(\log_2 n)$$

$$= 1 + \log_2(\log_2 n)$$

$$\boxed{T(n = O(\log_2(\log_2 n))}$$

$$T(n) = T\left(\left\lfloor \frac{2^k - 1}{2^{k-1}} \right\rfloor\right) + (k-1).$$

$$= T\left(\left\lfloor \frac{2^k}{2^{k-1}} - \frac{1}{2^{k-1}} \right\rfloor\right) + (k-1)$$

$$= T\left(\underbrace{\left\lfloor 2 - \frac{1}{2^{k-1}} \right\rfloor}_{\lfloor 2 - \text{something less than } 1 \rfloor = 1}\right) + (k-1)$$

$$T(n) = T(1) + (k-1) \quad — (4)$$

Given, $n = 2^k - 1$

$$n + 1 = 2^k$$

$$\boxed{k = \log_2(n+1)} \quad — (5)$$

Substituting (5) in eq(4),

we get,

$$T(n) = 1 + \log_2(n+1) - 1$$

$$T(n) = \log_2(n+1)$$

$$\boxed{T(n) = O(\log_2(n+1))}$$

4 (c) $T(n) = 3T\left(\frac{n}{2}\right) + 8n, \quad n = 2^k, T(1) = 1.$

$$T\left(\frac{n}{2}\right) = 3T\left(\frac{n}{2^2}\right) + \left(8 \times \frac{n}{2}\right).$$

$$\cancel{T\left(\frac{n}{2^2}\right) = 3}$$

$$T(n) = 3T\left(\frac{n}{2^2}\right) + 3 \cdot 8 \frac{n}{2} + 8n.$$

$$= 3^{\cancel{k}} T\left(\frac{n}{2^{\cancel{k}}}\right) + \sum_{i=0}^{\cancel{k}-1} \left(\frac{3}{2}\right)^i \times 8n$$

$$\frac{n}{2^r} = 1 \Rightarrow 2^k = 2^r \Rightarrow k = r.$$

$$= 3^k \cdot T\left(\frac{n}{2^k}\right) + \sum_{i=0}^{k-1} \left(\frac{3}{2}\right)^i \times 8n$$

$$= 3^{\log n}(1) + 8n \sum_{i=0}^{(k-1)} \left(\frac{3}{2}\right)^i.$$

From $i=0$ to $k-1$, Total 'k' terms are in the series.

$$S_n = \frac{a(r^n - 1)}{r-1}.$$

$r \to$ common ratio $= \frac{3}{2}$.

$a =$ first term.

$a = \left(\frac{3}{2}\right)^0 = 1.$

$$= \frac{1\left(\left(\frac{3}{2}\right)^{k} - 1\right)}{0.5}.$$

$$= 3^{\log n} + 16n\left[\left(\frac{3}{2}\right)^{\log n} - 1\right].$$

$$= \cancel{3}\, n^{\log_2 3} + 16n\left[(1.5)^{\log n} - 1\right]$$

$$= n^{\log_2 3} + 16n\left[n^{\log 1.5} - 1\right].$$

$$= n^{\log_2 3} + 16n\left[n^{0.58} - 1\right]$$

$$= n^{1.5849} + 16\, n^{1.5849} - 16n.$$

$$= O\left(n^{1.5849}\right).$$

5. (a) Apply Big O definition, try to find constant value in each case.

$$5^{n+1} = O(5^n) \checkmark$$

$$5^{5n} = O(5^n) \times.$$

(b) If $\log^* n = i$, then $\log^*(\log n) = i - 1$.

$\log(\log^* n) = \log i$.

So $\log^*(\log n)$ is asymptotically larger.

6 (a) Worst case time to sort a list of length 'z' by insertion sort is $\Theta(z^2)$.

$\left(\frac{n}{z}\right)$ sublists, each of length 'z' takes

$$\Theta\left(z^2 \cdot \frac{n}{z}\right) = \Theta(nz).$$

(b) $\frac{n}{z}$ sorted sublists each of length z.

To merge n/z sorted sublists to length 'n', we will take 2 sublists at a time and continue to merge. ~~This~~

$\log\left(\frac{n}{z}\right)$ steps and we compare n elements in each step.

$$\therefore \Theta\left(n \log\left(\frac{n}{z}\right)\right)$$

(c) Modified algorithm, $= \Theta\left(nz + n\log\left(\frac{n}{z}\right)\right)$

$= \Theta(n \log n)$.

lets say $z = \Theta \log n$.

$\Theta(nz + n\log(n/z)) = \Theta(nz + n\log n - n\log z)$

$= \Theta(n\log n + n\log n - n\log(\log n$

$= \Theta(n \log n)$.

Scanned with CamScanner

(d) We already found out Z value to be around '44' in HW 2, last problem.

$T(n) = \underline{\underline{8}}\, n^2 \rightarrow$ refers to insertion sort

$T(n) = \underline{\underline{64}}\, n \log n \rightarrow$ refers to merge sort

The constant factors in insertion sort make it faster in practice for small problem sizes on machines.

(e) Write the program in any language, which uses insertion sort to sort upto 44 elements and after that use merge sort.