19K-1135, Abdul-Rehman

Q1) (a) $T(n) = 5 + 1 + (n+1) + n + n$

$= 3n + 7$

Now, for

$T(n) \leq c f(n)$

$3n + 7 \leq c(n)$

$3 + 7/n \leq c$

if $n_0 = 1$, then

$\boxed{10 \leq C}$

(b) $T(n) = 3(\log_2 n)^2 + 2\log_2 n + 6$

Now, for

$T(n) \leq c f(n)$

$3(\log_2 n)^2 + 2\log_2 n + 6 \leq c(\log_2 n)^2$

$3 + \dfrac{2}{\log_2 n} + \dfrac{6}{(\log_2 n)^2} \leq c$

if $n_0 = 2$, then

$3 + \dfrac{2}{1} + \dfrac{6}{(1)^2} \leq c$

$\boxed{11 \leq c}$

(c) $T(n) = 8 + 1 + n + 1 + n + n + n + 2n + \frac{5n^2}{3}$

$\cancel{= 5n^2 + \frac{n^2}{3} + 10} = 6n + \frac{5n^2}{3} + 10$

Now, for

$T(n) \le c f(n)$

$6n + \frac{5n^2}{3} + 10 \le c(n^2)$

$\frac{6}{n} + \frac{5}{3} + \frac{10}{n^2} \le c$

where $n_0 = 1$

$\frac{(6 \times 3) + 5 + 30}{3} \le c$

$\frac{18 + 5 + 30}{3} \le c$

$\frac{53}{3} \le c$

(d) $T(n) = 7 + 1 + 3\log_2 n + 3n\log_2 n + 3n(\log_2 n)^2$

Now, for

$T(n) \le c f(n)$

$8 + 3\log_2 n + 3n\log_2 n + 3n(\log_2 n)^2 \le c(n(\log_2 n)^2)$

$\frac{8}{n(\log_2 n)^2} + \frac{3}{n\log_2 n} + \frac{3}{\log_2 n} + 3 \le c$

Scanned with CamScanner

where $n_0 = 2$

$$\frac{8}{2} + \frac{3}{2} + \frac{3}{1} + 3 \le c$$

$$\frac{8+3+6+6}{2} \le c$$

$$23/2 \le c$$

(e) $T(n) = 1 + \log_2 n + \frac{\log_2 n}{2} (\log_2 n + 1)$

Now, for

$$T(n) \le c\, f(n)$$

$$1 + \log_2 n + \frac{(\log_2 n)^2}{2} + \frac{\log_2 n}{2} \le c\, (\log_2 n)^2$$

$$\frac{1}{(\log_2 n)^2} + \frac{1}{\log_2 n} + \frac{1}{2} + \frac{1}{2(\log_2 n)} \le c$$

where $n_0 = 2$

$$1 + 1 + \frac{1}{2} + \frac{1}{2} \le c$$

$$\frac{2+2+1+1}{2} \le c$$

$$3 \le c$$

Q2) Insertion : $4n^2$, Merge : $32 n \log_2 n$

$$4n^2 \le 32 n \log_2 n$$

$$\frac{4n^2}{32n} \le \log_2 n$$

$$\frac{n}{8} \le \log_2 n$$

Taking both sides as power of 2

$$2^{n/8} \le 2^{\log_2 n}$$

$$2^{n/8} \le n$$

If we plot graph, we get result

$$2 \le n \le 43$$

for which the given condition holds.

Q3) 1^{st} Alogr : $100 n^2$
2^{nd} Alogr : 2^n

$100 n^2 \le 2^n$, by plotting graph
where let $n = 15$

$$22500 < 32768$$

So, $\boxed{n = 15}$

Q4) Given

$T(n) = \frac{1}{8}n^3 - 5n^2$

for $\Theta(n)$, we need

$\frac{1}{8}n^3 - 5n^2 \le c_1 n^3$ | $\frac{1}{8}n^3 - 5n^2 \ge c_2 n^3$

$\cancel{\frac{1}{8} - \frac{5}{n} \le c_1}$ | $\cancel{\frac{1}{8} - \frac{5}{n} \ge c_2}$

~~where $n_0 = 1$~~

$\cancel{\frac{1}{8} - 5 \le c_1}$

$\cancel{-\frac{39}{8} \le c_1}$

$\frac{1}{8}n^3 - 5n^2 \le c_1 n^3$ | $\frac{1}{8}n^3 - 5n^2 \ge c_2 n^3$

let $c_1 = \frac{1}{8}$ | $\frac{1}{8}n^3 - 5n^2 \ge \frac{1}{32}n^3$

So, $\frac{1}{8}n^3 - 5n^2 \le \frac{1}{8}n^3$ | So, $\frac{1}{8}n^3 - 5n^2 \ge \frac{1}{32}n^3$

least upper bound | Highest lower bound

$$\boxed{\frac{1}{32}n^3} \le \frac{1}{8}n^3 - 5n^2 \le \boxed{\frac{1}{8}n^3}$$

For, $\Theta(n^3)$.

Scanned with CamScanner

Q5) ~~void Selection Sort (A)~~

```
void Selection Sort (Arr[], ArrSize){
    int i, j, Smallest;
    for(i=0; i< ArrSize-1; i++){
     Smallest=i;
     for(j=i+1; j<n; j++){
       If(Arr[j]<Arr[Smallest]&&
            Smallest ≠ j){
          Smallest=j;
       Swap(Arr[i], Arr[Smallest]);}
          }};
```

Cases:

Best Case $= O(n^2)$, Worst Case $= O(n^2)$

Average Case $= O(n^2)$

Q6) For simplicity, merge is simply equal to sum of the list lengths.

So, for first merge, $T(1) = n+n$

for second merge, $T(2) = n+2n$

The overall sum of these numbers would be $m\sum_{t=2}^{p}$ $\frac{}{p}$ $= \frac{m(P(P+1)-1)}{2}$ $O(mp$

(b) It can be done in linear time $O(mP)$ if we compare the whole arrays at one time instead of sorting them separately.

~~Q7) (a) $T(n) = T(\frac{n}{2}) + O$ $T(n) = O(\log_2 y)$~~.

Q7) $T(n) = O(\log_2 y)$

Q8) (a) $T(n) = n^3 \sum_{i=0}^{\infty} (1/8)^i$

$$= n^3 \frac{1}{1-(1/8)}$$

$$= O(n^3)$$

(b) $T(n) = \sum_{i=0}^{\log_{5/4} n} 3^i$

$$= \frac{3^{\log_{5/4} n+1} - 1}{2}$$

$$= \frac{3(3^{\log_{5/4} n}) - 1}{2}$$

$$= O(\log n^4)$$

(c) $T(n) = \sum_{k=1}^{n} 1/k$

$= \log_2 n + O(1)$

$= O(\log_2 n)$

(d) $T(n) = \sum_{i=0}^{\log_2 n} \frac{n}{\log_2 n - \log_2 2^i}$

$= O(n \log_2 (\log_2 n))$

(e) $T(n) = \sum_{i=0}^{\log_2 n} 2$

$= \frac{2^{\log_2 n+1} - 1}{2}$

$= \frac{2n-1}{2}$

$= O(n)$

Scanned with CamScanner