BHAVYA SREE PYLA

MID-TERM - ALGORITHM

Ans1.) a) $0.01n^3 + 0.1n^2 + n + 10$

The dominant term above is $0.01n^3$

Θ-notation → $\Theta(n^3)$

Reasoning → the cubic term dominates as n grows larger, therefore the lower-order terms and constants i.e. $[0.1n^2 + n + 10]$ can be ignored where 10 → constant.

b) $n + \lg n$

Θ notation → $\Theta(n)$

Reasoning → The logn term in $n + \lg n$ is negligible compared the linear term n
∴ The linear term will dominate.

Q. $\binom{n}{2}$ (n choose 2)

$\Rightarrow$ 

can be simplified to $\frac{n(n-1)}{2}$

$$\Rightarrow \frac{n}{2} \times \frac{n-1}{2}$$

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

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

$$= \left(\frac{1}{2}\right)n^2 - \left(\frac{1}{2}\right)n$$

$\therefore$ Θ notation $\rightarrow \Theta(n^2)$

The quadratic term will obviously dominate the linear term.

$\Rightarrow$

d) $n^2 + O(n^2)$

Θ notation $\rightarrow \Theta(n^2)$

Reasoning $\rightarrow$ the dominating term is n^2, the $O(n^2)$ is also n^2, however this will not affect the notion it will still be $\underline{\Theta(n^2)}$

e) $n^2 + o(n^2)$

Θ notation $\rightarrow \Theta(n^2)$

Reasoning $\rightarrow$ In the above case, $O(n^2)$ does not affect the notation, in this case however n^2 grows strictly faster than $o(n^2)$ $\therefore$ it will remain $\underline{\Theta(n^2)}$

$\Rightarrow$

2) a) Insertion Sort checks with the left elements then inserts the key if needed.

LIST => [12, 10, 15, 19, 10, 7, 6, 9]

Key = 10 # the algorithm loop starts at i=2 :: key = A[2] = 10
# checks with left element then inserts 10 before 12

=> [10, 12, 15, 19, 10, 7, 6, 9]

=> [10, 12, 15, 19, 10, 7, 6, 9]

=> [10, 12, 15, 19, 10, 7, 6, 9]

=> [10, 10, 12, 15, 19, 7, 6, 9]

=> [7, 10, 10, 12, 15, 19, 6, 9]

=> [6, 7, 10, 10, 12, 15, 19, 9]

Sorted List =>
=> [6, 7, 9, 10, 10, 12, 15, 19]

b) Loop invariant

→ At the starting of each iteration of the first i.e outer-Loop the sub-array A[1:i-1] is sorted.

1) Initialization → True at the start of each loop
↳ when i=2, A[i] is sorted.

2) Maintenance → if true at the start of the loop, true at the end.
↳ each loop will insert A[i] into the right position in A[1:i-1]

3) Termination → loop terminates
↳ when i = n+1 loop terminates
given in the for loop n is counted.

⇒

c) Worst case running time

$\Theta(n^2)$

i) The inner loop might run $i-1$ times for each i in the loop when the array is completly reversed (sorting wise)

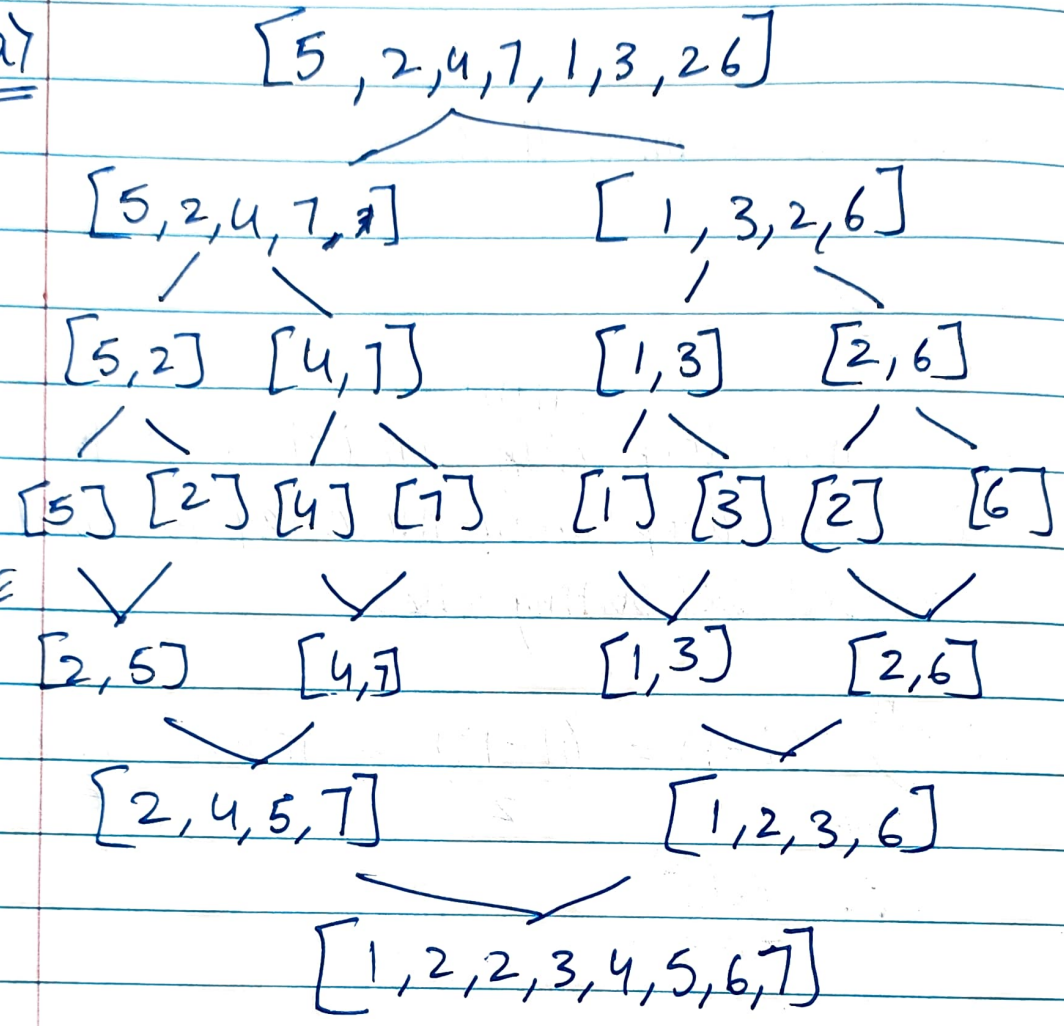
2) $\sum_{i=2}^{n} (i-1) = \frac{(n-1)n}{2} = \frac{n^2-n}{2} = \left(\frac{1}{2}\right)n^2 - \left(\frac{1}{2}\right)n$

$\therefore \Theta(n^2)$

not enough place for merge sort

PTO =>

3 a)



Sorted list => [1, 2, 2, 3, 4, 5, 6, 7]

b) Recurrences

$\frac{n}{2}$ → at each step split into $\frac{n}{2}$

$2T\left(\frac{n}{2}\right)$ → 2 recursive calls of the $\frac{n}{2}$

$\Theta(n)$ → merging takes linear time.

The base case will be $= \Theta(1)$

$$T(n) = \begin{cases} \Theta(1) & \text{if } n=1 \\ 2T(n/2) + \Theta(n) & n>1 \end{cases}$$

Solving the above recurrance using master method.

$$T(n) = 2T(n/2) + \Theta(n)$$

$a=2$; $b=2$; $f(n) = \Theta(n)$ ---①

watershed function $= n^{\log_b a}$ ---②

Substituting ① in ②

$n^{\log_b a} = n^{\log_2 2} = n^1$ ---③

Comparing ① and ③

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

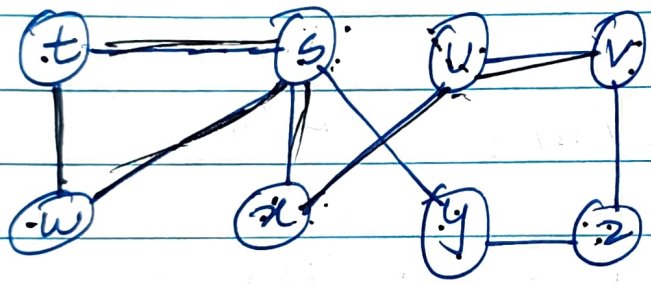
This is masters method case-2

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

$\Theta(n^1 \log n)$

$\Rightarrow \underline{\Theta(n \lg n)}$

4 a) a)



(for my case)

alphabetic order → [s, t, u, v, w, x, y, z]

BFS

1) starts at s
2) visit all neighbours of s in an alphabetic order → enqueue them
3) Then dequeue next vertex and visit the unvisited
4) continue until all of them are discovered.

Adjacenly List

s → t, v, w, x, y
t → s
v → s, v
v → v
w → s
x → s
y → s
z → y

EXECUTION

Queue	Discovered	Parent	Distance
s	s	NIL	0
t, v, w, x, y	t	s	1
v, w, x, y	v	s	1
w, x, y	w	s	1
x, y	x	s	1
y	y	s	1
v	v	v	2
z	z	y	2

⇒ s → t
s → v

$S \to w$

$S \to x$

$S \to y$

$v \to v$

$y \to z$

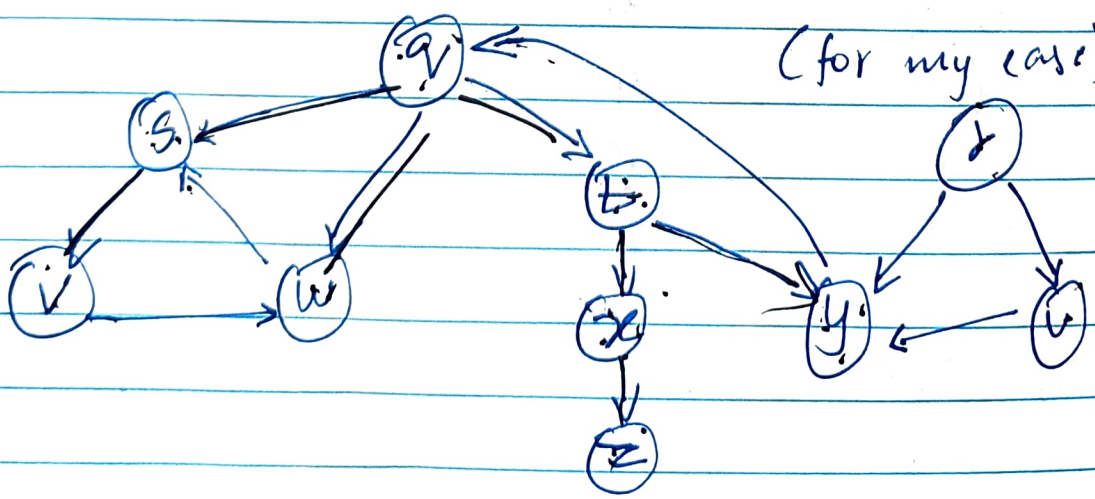
b) Running time.

→ Each vertex is enqueued or dequeued once
→ $\Theta(V)$

→ Each ~~vertex~~ edge is examined atleast once → $\Theta(E)$

Total → $\Theta(V+E)$
→ tight bound since both terms are required.

Ans 5)



Alphabetic order → (q, r, s, t, u, v, w, x, y, z)

Adjacency List

q : s, t
r : u
s : q, t, v, w
t : q, r, u, x
u : x, t
v : s, w
w : s, v
x : t, y
y : x, z
z : y

DFS

1) visit first vertex
2) mark discovery time
3) Explore ~~out~~ the neighbours vertically and mark discovery time.
4) Mark finish time when all adjacent vertices were explored.

EXECUTION

VERTEX	DISCOVERY	FINISH	PARENT	EDGE
q	1	16	-	
s	2	9	q	tree
v	3	6	s	tree
w	4	5	v	tree
t	7	15	s	tree
r	8	13	t	tree
u	9	12	r	tree
x	10	11	t	tree
y	14	15	x	tree
z	15	16	y	tree

Edge classification

1) Tree edges → $q \to s$, $s \to v$, $v \to w$, $s \to t$, $t \to r$, $r \to u$, $t \to x$, $x \to y$, $y \to z$

2) Back edges → it is a graph ∴ a tree without duplicates

3) forward edges → none

Time complexity → $\Theta(V+E)$