Design and Analysis of Algorithms

Tutorial - 7

Name - Riya Negi
Section - CE
Roll No. - 28
University Roll No - 2015543

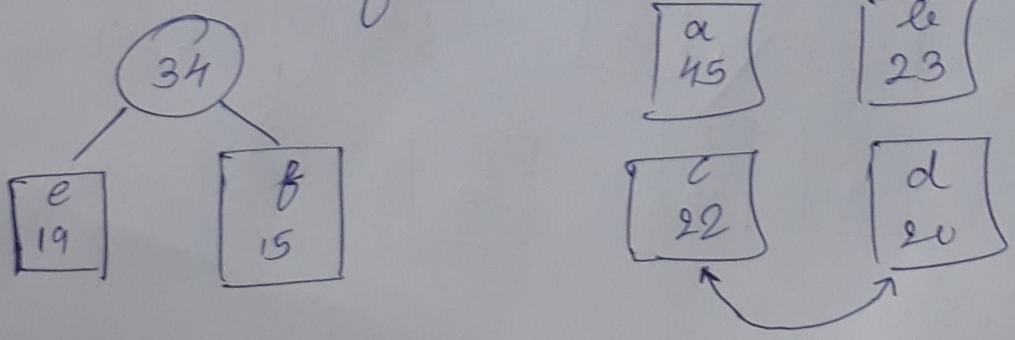
Answers

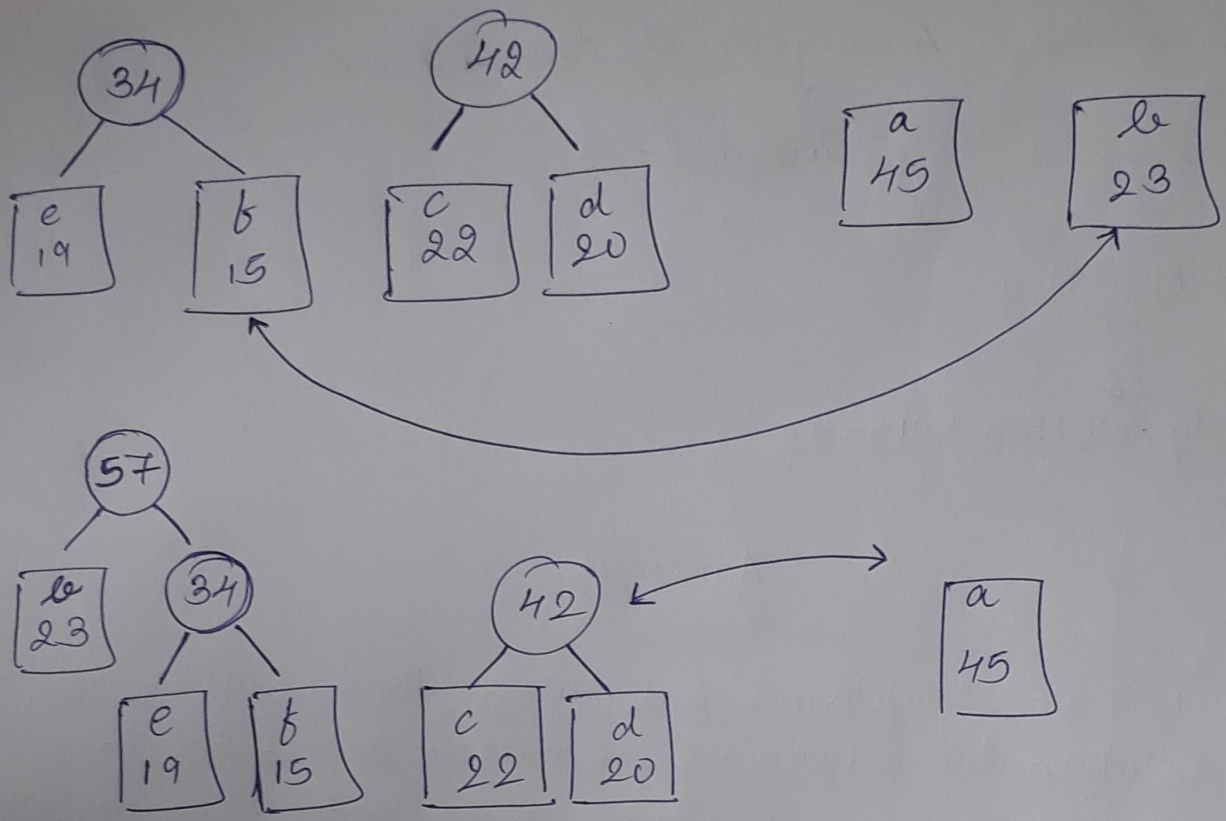
Ans 1 It is an algorithmic paradigm that builds up a solution by adjoining smaller pieces together, always choosing the next piece that offers the most obvious and immediate benefit.
We should use greedy approach whenever a locally optimal solution is also globally optimal

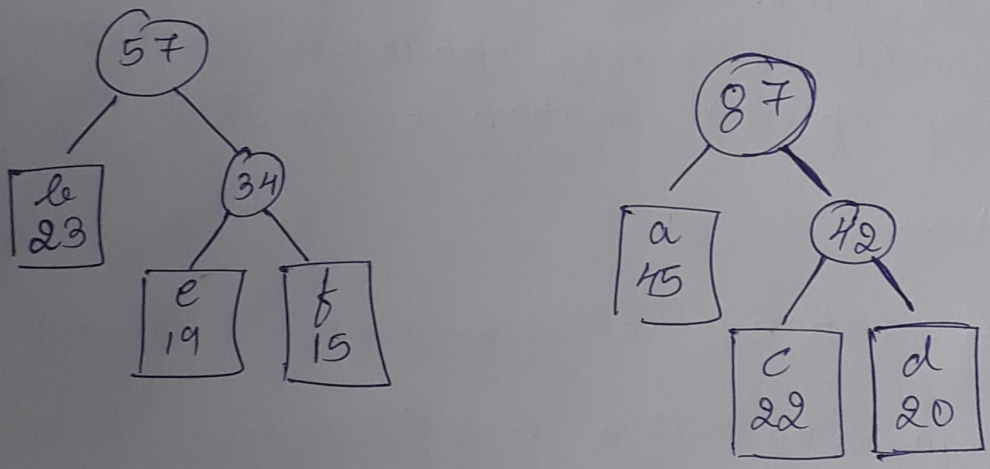
Ans 2

Name	TC	SC
Activity Selection	$O(n \log n) \leftrightarrow O(n)$	$O(n)$
Job Sequencing	$O(n^2) \leftrightarrow O(n \log n)$	$O(n)$
Fractional Knapsack	$O(n \log n) \leftrightarrow O(n)$	$O(n)$
Huffman Encoding	$O(n \log n) \leftrightarrow O(\log n)$	$O(n)$

Ans 3 a = 45 d = 20 b = 23 e = 19
c = 22 f = 15







Final Huffman Tree

a = 10
b = 00
c = 110
d = 111
e = 010
f = 011

Total bits used = $(45 \times 2) + (23 \times 2) + (22 \times 3) + (20 \times 3)$
$+ (19 \times 3) + (15 \times 3)$ = 364 bits

Ans 4 A 2-tree is used to implement Huffman encoding algorithm. It is a binary tree where every node has either 2-child or no child.

Applications of Huffman Encoding

- Data compression in long files without any loss
- To implement traffic routes with traffic magnitude

Ans 5

v	10	5	15	7	6	18	3
w	2	3	5	7	1	4	1
v/w	5	1.67	3	1	6	4.5	3

$k = 15 - 1 - 2 - 4 - 5 - 1 - 2 = 0$

Profit = $30 + 10 + 18 + 15 + 3 + 3.34$
= 79.34

v	6	10	18	15	3	5
w	1	2	4	5	1	3
v/w	6	5	4.5	3	3	1.67

Ans 6. Fractional Knapsack : It is using a greedy approach as we have divided our profits to the smallest unit possible & then builds upon it.

Huffman Encoding - It is using the greedy approach as we have divided our profits to the smallest unit possible & then builds upon it.

Huffman Encoding - It is using the greedy approach as it always places the node with the lower frequency further from the parent node.

Ans 7.

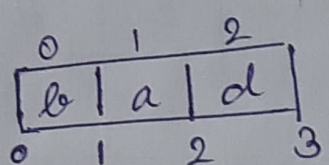
Start	1	2	0	6	9	10
End	3	5	7	8	11	12
Index	0	1	2	3	4	5

Jobs to do = [0], [3], [4] or [5]

i.e. => Max = 4

Ans 8

	Profit	Deadline
a	20	2
b	15	2
c	10	1
d	5	3
e	1	3



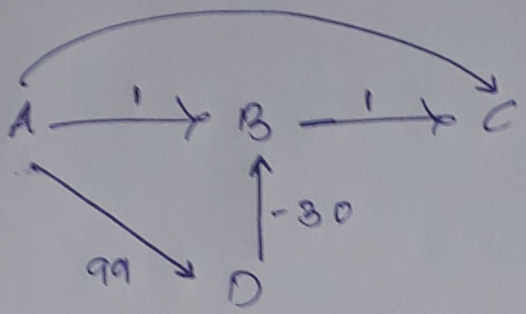
Profit = 20 + 15 + 5

= 40

Ans 9 Times when not to use greedy approach

- When the approach involves a lot of assumptions, a such as "pick always the ...."
- We should avoid greedy approach on complex implementation.
- When we are making performance-critical applications.

Eg. Dijkstra's algorithm is very unoptimized for graphs with negative edges.



We can't find the distance of the pair [A, C] → it gives 0, though it is −200

Ans 10 Normally, the time complexity of Job sequencing is $O(n^2)$ but we can improve it using a Priority Queue, made of Max Heap.

Algorithm

1. Sort the jobs based on deadlines.
2. Iterate the end & calculate the available slots between two consecutive deadlines include all data in Max-Heap.
3. If there are slots available & there are jobs in the Max Heap, include the job ID with max profit & deadlines in the result.

(4) Sort the array based on deadlines.

Time complexity : $O(n \log(n))$

Space complexity = $O(n)$