

Name:- Abhay Raneat

Course:- B.Tech (CSE)

Sec:- B

Sem:- 5

Roll No:- 01

University Roll No:- 1961002

Design & Analysis of Algorithms  
Assignment 7

Ques 1 What is the Greedy algorithmic paradigm?  
When should you make use of Greedy Algorithm in problem solving?

Ans Greedy is an algorithmic paradigm that builds up a solution piece by piece. always choosing the next piece that offers the most obvious & immediate benefits. So the problems where choosing locally optimal about also leads to global sol<sup>n</sup> are best fit for greedy.

For example = Knap Sack Problem

Greedy algorithm try to find the optimal sol<sup>n</sup> by taking the best available choice at every step.

Ques 2 Analyse the time & space complexity of the following algorithm.

(i) Activity selections

Space complexity =  $O(N)$

Time complexity

when activities are sorted by their finished time

$TC = O(n)$ .



when activities are not sorted by their finished time. the time complexity is  $O(n \log n)$  due to complexity of sorting

(ii) Job sequencing

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

Time complexity of job sequencing with deadline =  $O(n^2)$

Space complexity =  $O(n)$

(iii) Fractional Knapsack

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

Time complexity using greedy algo =  $O(n^2)$

Space complexity =  $O(w)$   
↳  $w = \text{weight}$

(iv) Huffman Encoding :-

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

Space complexity =  $O(n)$

Ques 3 A file contains the following characters & their corresponding frequencies as shown below:-

a: 45, b: 23, c: 22, d: 20, e: 19, f: 15

We use Huffman coding for data compression, generate the encoding for a, b, c, d, e, f using Huffman encoding & find the average length of a character after compression.

Ans

$f = 15$

$e = 19$

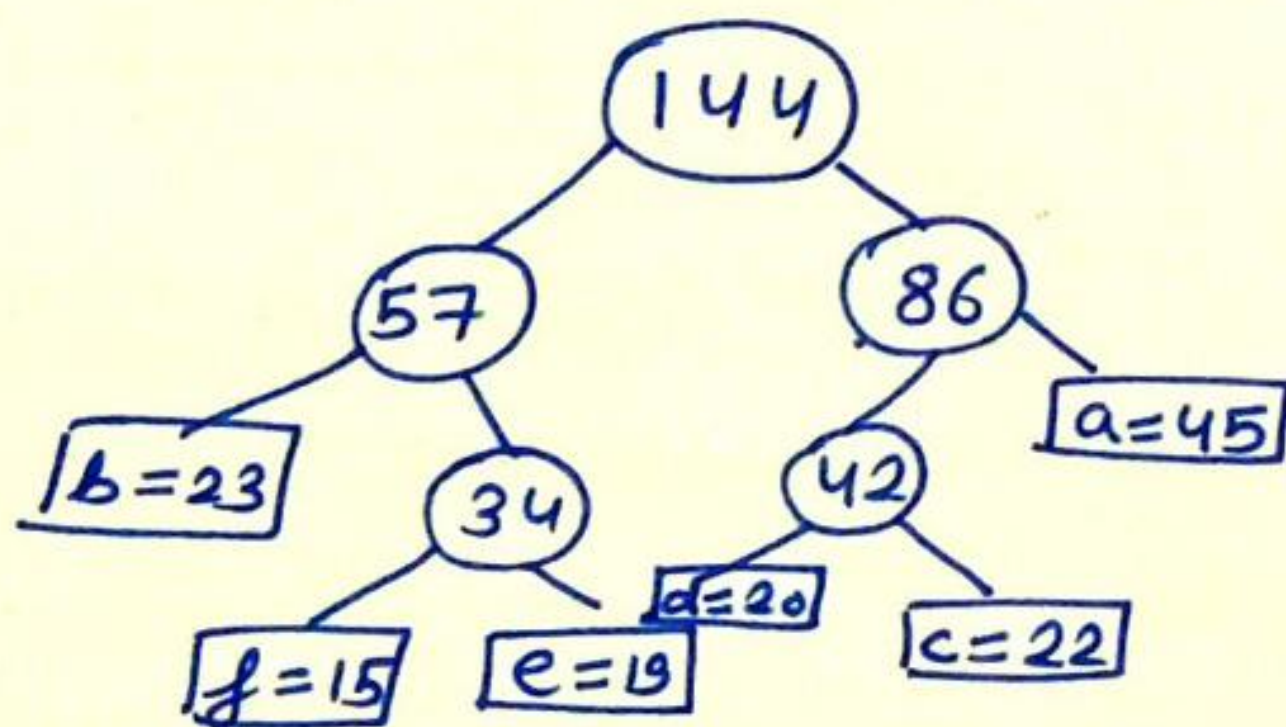
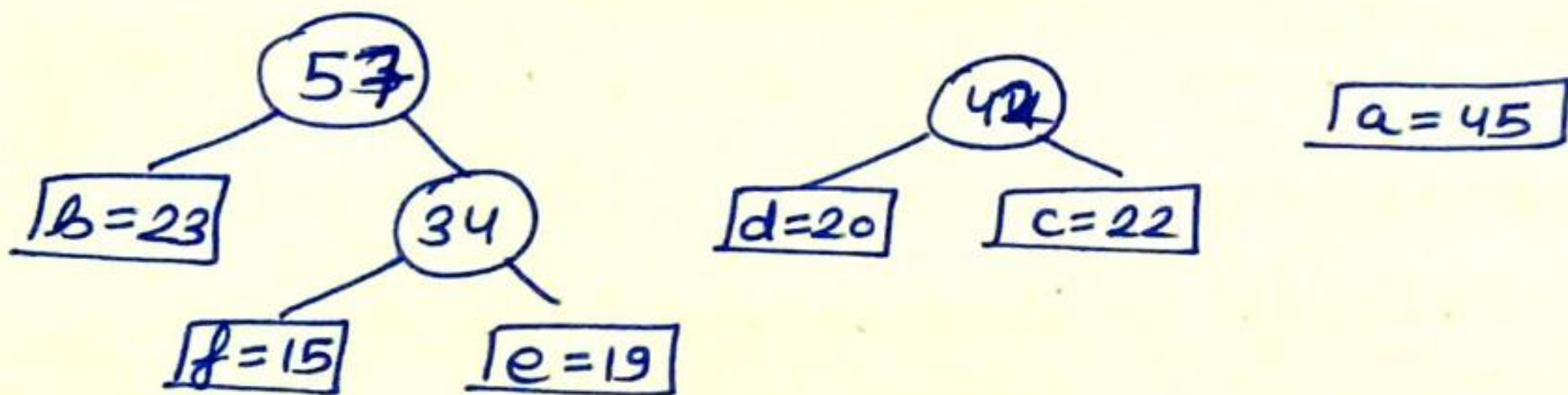
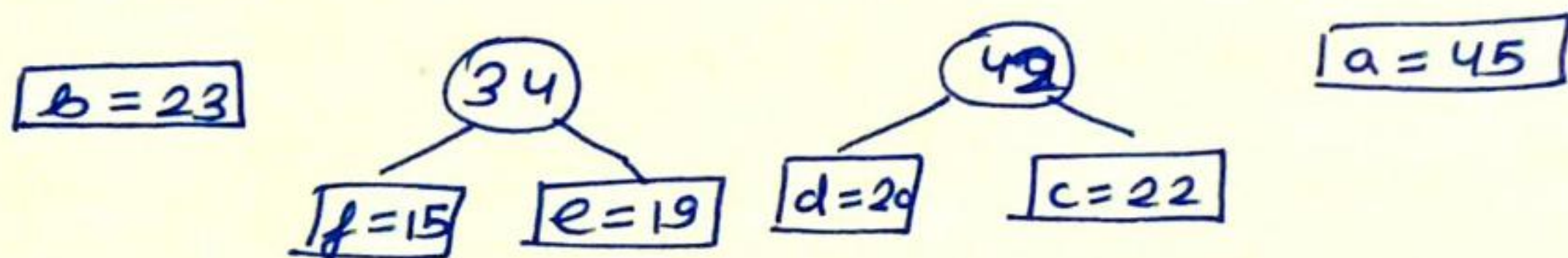
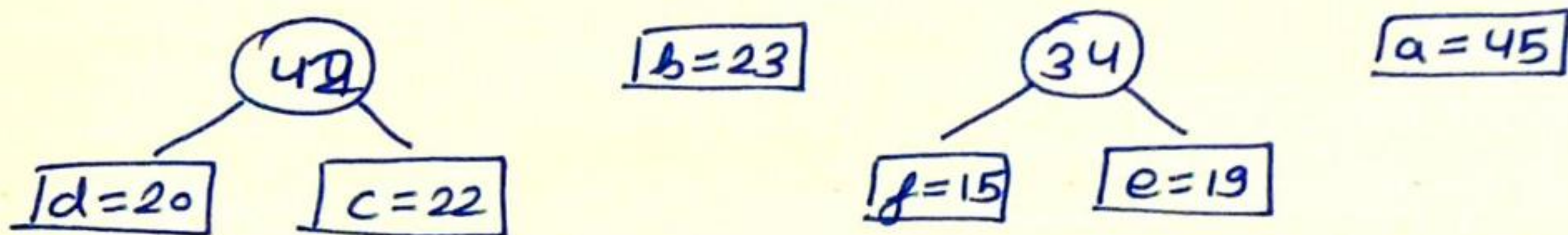
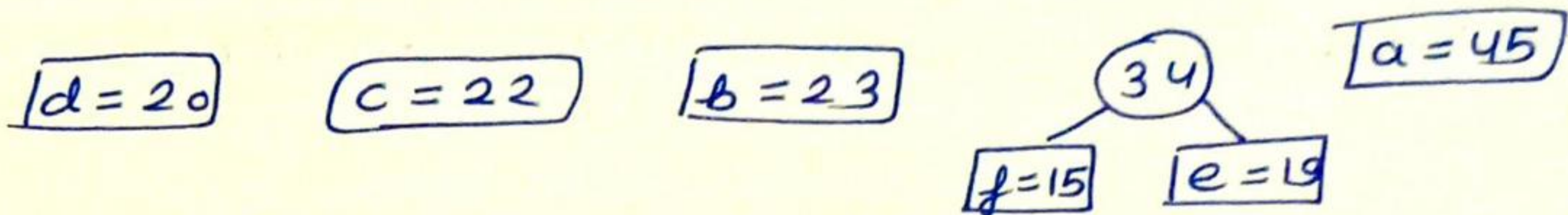
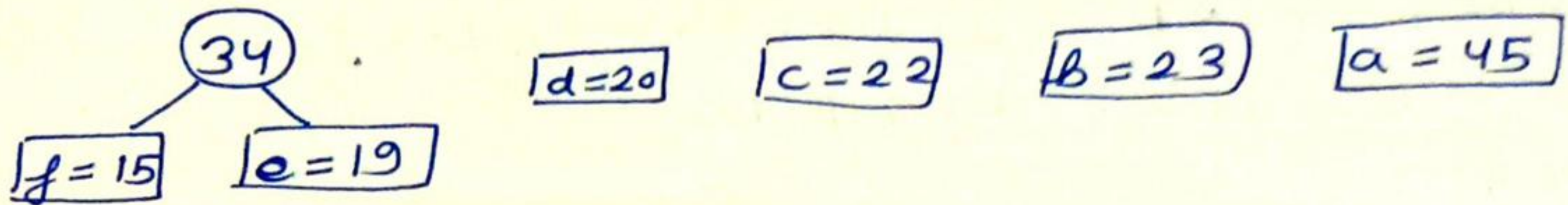
$d = 20$

$c = 22$

$b = 23$

$a = 45$





$b = 00$   
 $f = 010$   
 $e = 011$   
 $d = 100$   
 $c = 101$   
 $a = \phi 1$



$$\begin{aligned}
 &= 2 \times 45 + 2 \times 23 + 3 \times 22 + 3 \times 20 + 3 \times 19 + 3 \times 15 \\
 &= 90 + 46 + 66 + 60 + 57 + 45 \\
 &= 364
 \end{aligned}$$

$$\text{average total length} = \frac{364}{144} = 2.527778$$

Ques 4 Which data structure is used while implementation Huffman Encoding? What are the application of Huffman Encoding?

Ans Priority Queue is used for building the Huffman tree such that nodes with lowest frequency have the highest priority. A min heap data structure can be used to implement the functionality of a priority queue.

Application :- Huffman is widely used in all the mainstream compression formats that you might encounter. from GZIP, PKZIP & BZIP2 to image formats such as JPEG & PNG.

Ques 5 Given weights & values of 7 items, put these items in a knapsack of capacity  $W=15$  such that you get the maximum total value in the knapsack.

Value	10	5	15	7	6	18	3
Weight	2	3	5	7	1	4	1

Ans

Object	Value	Weight	fraction
1	10	2	$10/2 = 5$
2	5	3	$5/3 = 1.66$
3	15	5	$15/5 = 3$
4	7	7	$7/7 = 1$
5	6	1	$6/1 = 6$



6	18	4	$18/4 = 4.5$
7	3	1	$3/1 = 3$

$$\text{Weight} = 6 + 10 + 18 + 15 + 3 + 6 = \underline{\underline{58}}$$

Ques 6 Prove that Fractional Knapsack problem & Huffman Encoding has the greedy-choice property. (You need to provide proper explanation with an example as to why are these two algorithms categorized as greedy).

Ans In knap-sack problem, we used greedy approach bcz it is optimal approach if we use brute-force sol<sup>n</sup> would be to try all possible subset with all diff fm but that will be too much time taking & in knapsack we break item for maximizing the total values of knapsack.

object	1	2	3
value	24	27	30
weight	18	19	20

Sort the values/weight

$$\frac{30}{20}, \frac{27}{19}, \frac{24}{18}$$

$$30 + \frac{5}{19} \times 27 \times 20 = \underline{\underline{120}}$$

& in knapsack problem this is best & optimal none of other algorithms give like this answers.

### Huffman code

Greedy algorithm is an algorithm that follows the problem solving mechanism of making the locally optimal sol<sup>n</sup> at each stage with thinking of finding



a global optimum sol<sup>n</sup> for the problem & an Huffman coding in every stage ~~also~~ we try to find the prefix free binary code & minimize expected code word for optimum sol<sup>n</sup> for compress the data