Name - Dhruv Saini
Section - ML

Roll NO - 2015241, 24
Subject - Design & Analysis of Algo.
TCS 505

Tutorial

1 Greedy is an algorithmic paradigm that builds up a solution piece, always choosing the next piece that offers the most obvious and immediate benefit. This means that it makes a locally optimal choice in the hope that this choice will lead to a globally optimal solution.

A problem must comprise these 2 components for a greedy algorithm to work -

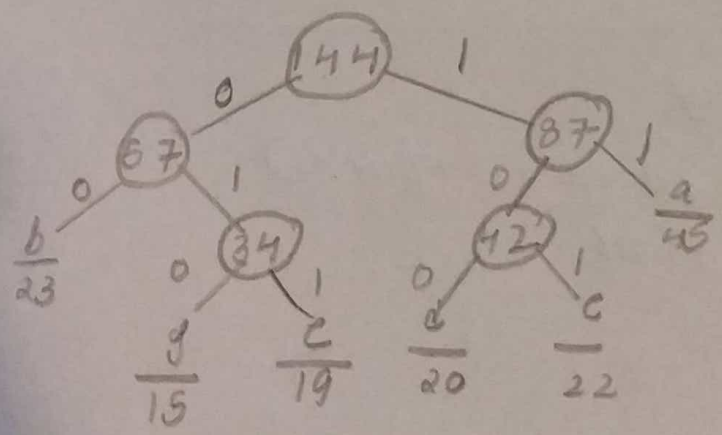
① It has optimal substructures. The optimal solution for the problem contains optimal solutions to the subproblems.

② It has a greedy property. If you make a choice that seems the best at that moment & solve the remaining sub-problems later, you still reach an optimal solution. You'll never have to reconsider your earlier choices

2

Algorithm	Time complexity	Space complexity
Activity selection	O(n log n) unsorted O(n) sorted	O(1)
Job sequencing	$O(n^2)$	O(n)
Fractional knapsack	O(n log n)	O(1)
Huffman encoding	O(n log n)	O(n)

3



Letters	Huffman code	Frequency	No of bits
a	00	45	90
b	11	23	46
c	010	22	66
d	011	20	60
e	100	19	57
f	101	15	45
		144	364

$$\text{Avg length} = \frac{364}{144} = 2.52$$

4

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.

Applications -

- Huffman encoding is widely used in compression formats like GZIP.
- Multimedia codecs like JPEG, PNG & MP3 use huffman encoding
- Huffman encoding is used for transmitting fax and text.

5

value	10	5	15	7	6	18	3
weight	2	3	5	7	1	4	1
w/v	6	5	4.5	3	3	1.11	1

capacity 15 kg

$$\text{weight} = 6 + 10 + 18 + 15 + 3 + (1.66 + 2)$$
$$= 55.33$$

6 • In fractional knapsack problem, the basic idea of the greedy approach is to calculate the ratio value/weight for each item & sort the items on the basis of this ratio. Then take the item with the highest ratio & add them until we can't add the next item as a whole & at the end, add the next item as much as we can.

• In Huffman Encoding, the algorithm builds the tree 'T' analogous to the optimal code in a bottom-up manner. It starts with a set of |C| leaves (C 'C' is the no of characters) and perform 'merging' operations to create the final tree.
Huffman's greedy algorithm uses a table of the frequencies of each character to build up an optimal way of representing each character as a binary string.

7 Sorted activities

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

↑ ↑ ↑

Max activities = 3

8

	A	B	C	D	E
Profit	20	15	10	5	1
Deadline	2	2	1	3	3

∴ Max Deadline = 3

Max Array size = 3

Job scheduling

C	A	D
0	1	2	3

Deadline →

Max. total profit = 10 + 20 + 5
= 35

9

- Sometimes greedy algorithm fail to find the globally optimal solution because they don't consider all the data. The choice make by a greedy algorithm may depend on choices it has made so far, but it is not aware of future choice it could make.

  Ex- Let us consider that the capacity of a knapsack is 25 (w = 25) & the items are as shown in the following table:

	A	B	C	D
Profit	24	18	18	10
weight	24	10	10	7

without considering the profit per unit weight if we apply greedy approach to solve this problem, first item 'A' will be selected as it'll contribute maximum profit among all the elements. After selecting 'A', no more item will be selected. Hence, for this given set of items, total profit is 24, whereas the optimal solution can be achieved by selecting items 'B' and 'C' where the total profit is 36.

10 we can optimise the approach of solving Job' sequencing problem by using Priority Queue (Max Heap).

Algorithm

① Sort the jobs based on their deadlines.

② Iterate from the end & calculate the available slots b/w every two consecutive deadlines. Include the profit, deadline and job ID of ith job in the max heap.

③ while the slots are available and there are jobs left in the max heap, include the Job ID with maximum profit & deadline in the result.

④ Sort the result array based on their deadlines