## Tutorial 7

**Q1.** What is the Greedy Algorithmic paradigm when should you should make use of Greedy Algorithms in problem solving?

→ Greedy Algorithm always makes the choice that seems to be the best at that moment. It makes a locally optimal choice in the hope that this choice will lead to a globally - optimal solution.

Used for optimization problems. (maximization or minimization)

A problem should be solved in stages, as we get input we see it that can be fit in result.

**Q2.** Analyse the time and space complexity of following Algorithms -

① Activity selection -
$TC = O(n \log n)$

② Job Sequencing
$TC = O(n^2)$
$SC = O(n)$

③ Fractional Knapsack
$TC = O(n \log n)$
$SC = O(n)$

Huffman Encoding — $O(n\log n)$

Q3. A file contains the following characters and their corresponding frequencies as shown below.

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

$f = 15 \times$
$e = +5\ 19 \times$
$d = 20 \times$
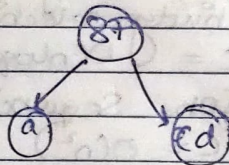$c = 22 \times$
$b = 23 \times$
$a = 245$
$ef = 34 \times$
$cd = 42$
$bef = 57$
$acd = 87$
$abcdef = 144$
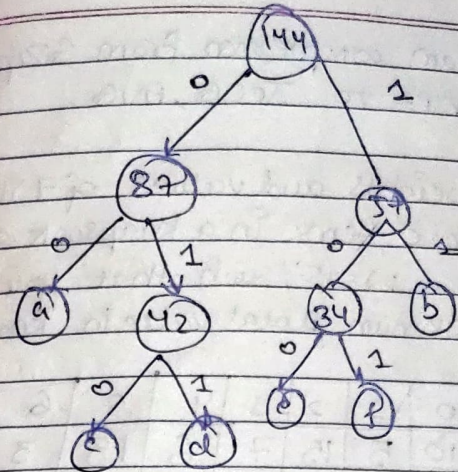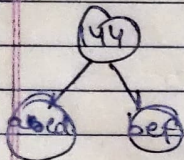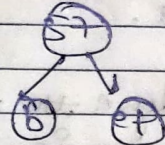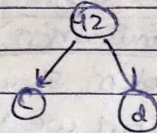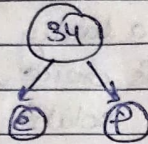


a = 00      c = 010     e = 100
b = 11      d = 011     f = 101

$a \rightarrow 2 \times 45 = 90$
$b \rightarrow 2 \times 23 = 46$
$c \rightarrow 3 \times 22 = 66$
$d \rightarrow 3 \times 20 = 60$
$e \rightarrow 3 \times 19 = 57$
$f \rightarrow 3 \times 15 = 45$
$\overline{\qquad 364 \qquad}$

Avg code length = $\dfrac{364}{144}$

Q4. Which data structure is used while implementing Huffman Encoding? Applicat. of Huffman's Encoding.
Heap (Min) is used for Huffman Encoding

→ Mainstream compression from Gzip/kzip and BZIP2 to JPEG, PNG.

**Q5** Given weights and values of 7 items, put these items in a knapsack of capacity W=15, such that you get the maximum total value in knapsack.

| index | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|-------|---|---|---|---|---|---|---|
| Value | 10 | 5 | 15 | 7 | 6 | 18 | 3 |
| Weight | 2 | 3 | 5 | 7 | 1 | 4 | 1 |
| V/W | 5 | 5/3 | 3 | 1 | 6 | 18/4 | 3 |
| | | 1.6 | | | 4.5 | | |

① Take 6
$$p = 6$$
$$K = 15 - 1$$
$$= 14$$

② Take 5
$$P = 6 + 10 = 16$$
$$K = 14 - 2$$
$$= 12$$

③ Take 4.5
$$P = 16 + 18 = 34$$
$$k = 12 - 4 = 8$$

④ Take 3
$$P = 34 + 15$$
$$= 49$$
$$k = 8 - 5 = 3$$

⑤ Take 3
$$P = 49 + 3$$
$$= 52$$
$$k = 3 - 1 = 2$$

⑥ Take 1.6 $= 52 + 2 \times \dfrac{5}{3} = 3.3 + 52$
$$P = 55.3$$

**Q6** Prove that Fractional Knapsack problem and huffman encoding has greedy-choice property.

In Fractional Knapsack, we choose the maximum value/weight value which is the best solution for attaining maximum profit in a given capacity. (at every step)

Also in huffman encoding we choose the the two minimum according to our greedy condition so that it can be easily compressed.
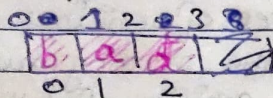
**Q7** Consider a set of activities given below, along with the starting and finishing time of each activity, find the maximum number of activities performed by a single person can only work on a single activity at a time.

| | 0 | 1 | 2 | 3 | 4 | 5 |
|-----------|---|---|---|---|---|----|
| Start time | 1 | 2 | 0 | 6 | 9 | 10 |
| End time | 3 | 5 | 7 | 8 | 11 | 12 |

= 3 activities.
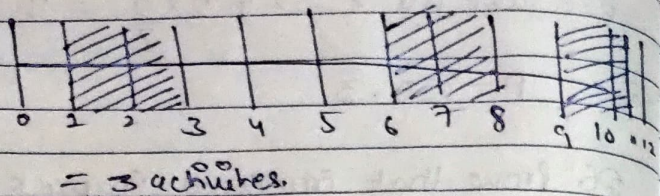
Q8 consider the following jobs where every job has a deadline and an associated profit if completed within deadline. It is given that every job takes a single unit of time. Perform job sequencing Such that you maximize the total profit if only one job can be scheduled at a time.

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

0 1 2 3

| b | a | d | |
|---|---|---|---|

0 1 2

① Pick 'a'     (20)
② Act→b '(15)     Profit = 20
③ Drop 'c'        +15
④ Pick d     (5)     40
                +15
                  40
                (5)
                ‾‾
                40  (Total profit)

Q9 Why should we avoid making use of greedy approach in problem solving? Give examples to support

Dijkstra's Algo with -ve weights. —
  if we decrease the value of edge, we will obtain a distance even further from actual minimum distance.

Travelling Salesman Problem. — greedy Strategy will find the worst possible Solutions.

Q10 How can you optimize the approach used to solve job sequencing problem?
— Using Dynamic Programming by nullifying the approach of sorting.

— Time complexity reduces from $O(n^2)$ to $O(m \times n)$