

Course/ Batch: BTech/ SCSET
Course Code: CSET244
Semester: Even
Session: 2024-2025

Course Type: Core
Course Name: Design and Analysis of Algorithms

Tutorial Assignment: Week 6

Tutorial title: Greedy method

Q1. We have given a fractional Knapsack problem with a maximum weight capacity of 11 kg. A greedy method sorts the items by their value-to-weight ratios in descending order and packs them greedily, starting from the first item in the ordered list.

Step 1: Compute Value-to-Weight Ratios

We first compute the value-to-weight ratio for each item:

Item Number	Weights (Kg.)	Values (in Rs.)	Value/Weight
1	10	60	6.0
2	7	28	4.0
3	4	20	5.0
4	2	24	12.0
5	8	48	6.0

Step 2: Sorting items in decreasing order of value per weight:

1. Item 4 $\rightarrow$ 12.00 Rs/Kg
2. Item 1 $\rightarrow$ 6.00 Rs/Kg
3. Item 5 $\rightarrow$ 6.00 Rs/Kg
4. Item 3 $\rightarrow$ 5.00 Rs/Kg
5. Item 2 $\rightarrow$ 4.00 Rs/Kg

Step 3: Pick Items Greedily

We have a knapsack capacity of 11 Kg.

Pick Item 4 (2 Kg, 24 Rs) $\rightarrow$ Remaining capacity: $11 - 2 = 9$ Kg

Pick Item 1 (10 Kg, 60 Rs), but only take 9 Kg $\rightarrow$ Remaining capacity: 0 Kg

Since Item 1's total weight is 10 Kg, and we need only 9 Kg:

Fraction taken $= 9/10 = 0.9$

Value taken $= 0.9 \times 60 = 54$

Step 4: Total value in the knapsack: Rs 24 + Rs 54 = Rs 78

Q2. Step 1: The given message is “MISSISSIPPI”, which consists of 11 characters.

Character	Frequency
M	1
I	4
S	4
P	2

Step 2: Sort characters by frequency:

M (1), P (2), S (4), I (4)

Merge the lowest frequency nodes iteratively:

Merge M (1) and P (2) → New node (MP: 3)

Merge MP (3) and one of (S or I) (4) → New node (MP-S: 7)

Merge MP-S (7) and I (4) → Root node (MP-S-I: 11)

Assign binary codes (example assignment):

I = 0

S = 10

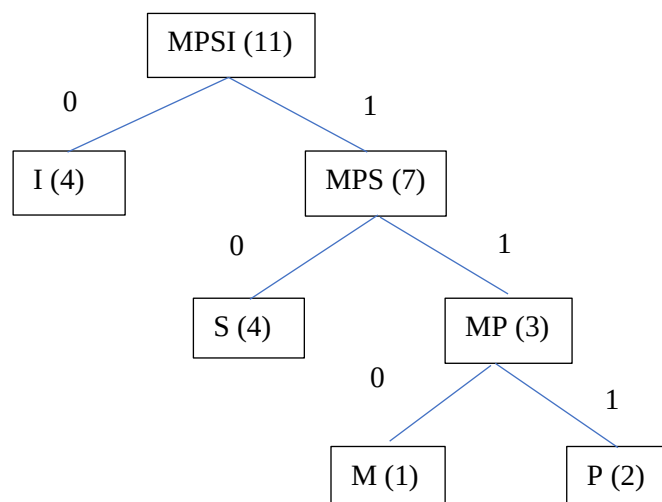
MP = 11 (further split into M = 110, P = 111)

Step 3:

Character	Frequency	Huffman Code	Bits Used
M	1	110	3
I	4	0	4
S	4	10	8
P	2	111	6

Total Bits Used:

$$(1 \times 3) + (4 \times 1) + (4 \times 2) + (2 \times 3) = 3 + 4 + 8 + 6 = 21$$



ii) Given encoded string: 100010111001010

1. $10 \rightarrow S$
2. $0 \rightarrow I$
3. $0 \rightarrow I$
4. $10 \rightarrow S$
5. $111 \rightarrow P$
6. $0 \rightarrow I$
7. $110 \rightarrow M$
8. $10 \rightarrow S$

Thus, the decoded message is: SIISPIMS

iii) The time complexity of constructing an optimal code using Huffman's algorithm is $O(n \log n)$, where n represents the number of unique characters in the input string.

Q3. Using greedy method for the activity selection problem, need minimum 4 classrooms.
After sorted by Finish time, we can greedily schedule classes into classrooms.

$$S_i = (1, 3, 2, 4, 8, 7, 9, 11, 9, 12)$$

$$F_i = (3, 4, 5, 7, 9, 10, 11, 12, 13, 14)$$

(1, 3), (4, 7), (8, 9), (11, 12) can be scheduled together.

(3, 4), (7, 10), (12, 14) can be scheduled together.

(2, 5), (9, 11) can be scheduled together.

(9, 13) can be scheduled alone.

Q4. Start at A

Nearest city: C (Distance = 15)

From C

Nearest unvisited city: D (Distance = 20)

From D

Nearest unvisited city: B (Distance = 35)

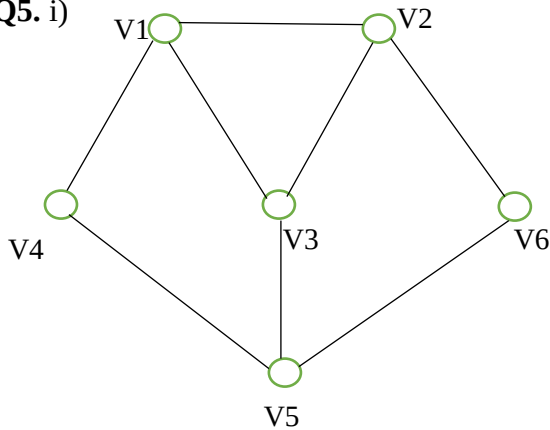
From B

Return to A (Distance = 20)

Final Tour Sequence: $A \rightarrow C \rightarrow D \rightarrow B \rightarrow A$

Total Distance Traveled: $15(A \rightarrow C) + 20(C \rightarrow D) + 35(D \rightarrow B) + 20(B \rightarrow A) = 90$

Q5. i)



Step1: Sort the vertices according to their descending order of degree: $\{(V1, 3), (V2, 3), (V3, 3), (V5, 3), (V4, 2), (V6, 2)\}$

Step2: $C = \{c1, c2, c3, \dots\}$

Pass 1

Step3: Assign $c1$ to $V1$ (as $c1$ is the first color of the color set C , $V1$ is the first vertex among the vertex Set V).

Step4: The remaining part of the ordered list V is traversed, and the same color $c1$ is allocated to vertex $V5$ for which no adjacent vertex has the same color.

Pass 2

Step3: Assign $c2$ to $V2$ (as $c2$ is the next color of the color set C , $V2$ is the first vertex among the vertex Set V which is not colored).

Step4: The remaining part of the ordered list V is traversed, and the same color $c2$ is allocated to vertex $V4$ for which no adjacent vertex has the same color.

Pass 3

Step3: Assign $c3$ to $V3$ (as $c3$ is the next color of the color set C , $V3$ is the first vertex among the vertex Set V which is not colored).

Step4: The remaining part of the ordered list V is traversed, and the same color $c3$ is allocated to vertex $V6$ for which no adjacent vertex has the same color.

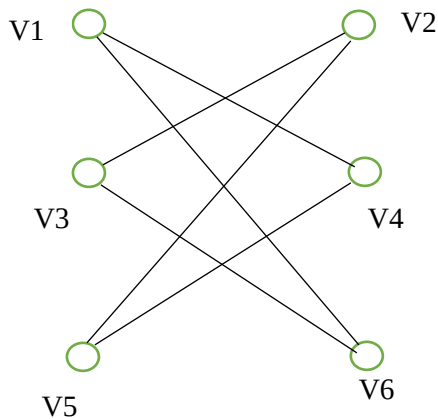
$\{V1, V5\} \rightarrow c1$

$\{V2, V4\} \rightarrow c2$

$\{V3, V6\} \rightarrow c3$

Three colors are required to color the given graph using Greedy heuristic so that, no adjacent vertex has the same color.

ii)



Step1: Sort the vertices according to their descending order of degree: $\{(V1, 2), (V2, 2), (V3, 2), (V4, 2), (V5, 2), (V6, 2)\}$

Step2: $C = \{c1, c2, c3, \dots\}$

Pass 1

Step3: Assign $c1$ to $V1$ (as $c1$ is the first color of the color set C , $V1$ is the first vertex among the vertex Set V).

Step4: The remaining part of the ordered list V is traversed, and the same color $c1$ is allocated to vertex $V2$ for which no adjacent vertex has the same color.

Pass 2

Step3: Assign $c2$ to $V3$ (as $c2$ is the next color of the color set C , $V3$ is the first vertex among the vertex Set V which is not colored).

Step4: The remaining part of the ordered list V is traversed, and the same color $c2$ is allocated to vertex $V4$ for which no adjacent vertex has the same color.

Pass 3

Step3: Assign $c3$ to $V5$ (as $c3$ is the next color of the color set C , $V5$ is the first vertex among the vertex Set V which is not colored).

Step4: The remaining part of the ordered list V is traversed, and the same color $c3$ is allocated to vertex $V6$ for which no adjacent vertex has the same color.

$\{V1, V2\} \rightarrow c1$

$\{V3, V4\} \rightarrow c2$

$\{V5, V6\} \rightarrow c3$

Three colors are required to color the given graph using Greedy Algorithm so that, no adjacent vertex has the same color.

The Greedy heuristic not always gives the optimal solution as in this case optimal solution is 2

$\{V1, V3, V5\} \rightarrow c1$

$\{V2, V4, V6\} \rightarrow c2$