

Pseudocode Outline: Topic 7

By Joseph Abraham

Lecture Scheduling Using Graph Coloring

BEGIN

Initialize input:

- Define inputString = "0110100101101011010110011100001101111000011100"
 - Set SIZE = 7 to reflect a 7×7 adjacency matrix
 - Create a 2D matrix adjMatrix[SIZE][SIZE] and parse the string into it row by row
-

Function: Parse the matrix

FOR i from 0 to SIZE-1:

 FOR j from 0 to SIZE-1:

 Set adjMatrix[i][j] = inputString.charAt(i * SIZE + j) - '0'

Function: Graph Coloring (Greedy Algorithm)

Initialize array colors[SIZE] to -1

Set colors[0] = 0

FOR each node u from 1 to SIZE-1:

 Mark all colors as available

 FOR each neighbor v:

 IF adjMatrix[u][v] == 1 AND colors[v] != -1:

 Mark colors[v] as unavailable

Assign the **lowest available color** to colors[u]

Return maxColor + 1 as the **minimum number of time slots**

Function: Count Unique Conflicts

Initialize conflictCount = 0

FOR i from 0 to SIZE-1:

 FOR j from i+1 to SIZE-1:

 IF adjMatrix[i][j] == 1:

 Increment conflictCount

Return conflictCount as **minimum number of students with conflicts**

Output:

- Print the adjacency matrix
 - Print result from graph coloring: **minimum number of lecture times**
 - Print result from conflict counter: **minimum number of unique conflicts**
-

END