Diagram

Description automatically generated

The discovery time and running time for these nodes are :

M(1,20)

N(21,26)

O(22,25)

P: (27, 28)

Q(2,5)

R(6,19)

S(23,24)

T(3,4)

U(7,8)

V(10,17)

W(11,14)

X(15,16)

Y(9,18)

Z(12,13)

B) the returned linked\_list after topological sort would be : p, n, o, s, m, r, y, v, x, w, z, u, q, t

Diagram

Description automatically generated

Running time and discovery time would be:

A(1,14)

B(3,4)

C(2,11)

D(5,19)

E(6,9)

F(7,8)

G(12,13)

Diagram

Description automatically generated

Answers:

a)

q r s t u v w x y z

discovery 1 17 2 8 18 3 4 9 13 10

finish 16 20 7 15 19 6 5 12 14 11

b)

Diagram

Description automatically generated

The red mark are the DFS tree, there are five different tree in the DFS forest.

c):

components: {r} -> {u} -> {q, y, t} -> {x, z} -> {s, w, v}

A picture containing table

Description automatically generated

In this solution, we could using the presume algorithm in the dynamic programming.

Here I set (a, b ) = (row1, col1)

And (C,d) = (row2, col2)

Code in python:

presum = [[0 for \_ in range(len(matrix[0])+ 1)] for i in range(len(matrix) + 1)]

if len(matrix[0]) == 0 or len(matrix) == 0:

return

for i in range(len(matrix)):

for j in range(len(matrix[0])):

presum[i + 1][j + 1] = presum[i+1][j] + presum[i][j+1] + matrix[i][j] - presum[i][j]

result = presum[row2 + 1][col2 + 1] - presum[row1][col2 + 1] - presum[row2 + 1][col1] + presum[row1][col1]

return result

The time and space complexity would be both O(1)