Recursion – Spot Question

Consider an ordering of numbers from ‘i’ to ‘j’ ($1 \le i, j \le n$) with ‘i’ as the first number and ‘j’ as the last number. The intermediate numbers (if any) in the ordering can be obtained from an n×n matrix S, containing numbers ranging from 1 to n, by recursively finding the successor of the first number till the successor is the last number. The successor of ‘i’ in the ordering of numbers from ‘i’ to ‘j’ is:

$$successor(i, j) = \begin{cases} NIL, & S[i, j] = j \text{ } or \text{ } S[i, j] = k, k \text{ } is \text{ } in \text{ } the \text{ } ordering \text{ } identified \text{ } so \text{ } far \text{ } from \text{ } the \text{ } start \\ S[i, j], & S[i, j] \ne j \text{ } and \text{ } S[i, j] \ne k, for \text{ } all \text{ } k \text{ } in \text{ } the \text{ } ordering \text{ } identified \text{ } so \text{ } far \text{ } from \text{ } the \text{ } start \end{cases}$$

Here, S[i,j] = j indicates that we have determined the ordering and S[i,j] = k indicates that there is no valid ordering from ‘i’ to ‘j’. Given S, i and j, implement an algorithm to determine the ordering from ‘i’ to ‘j’.

Function Prototype: `int* ordering(int **S, int i, int j);`

Sample:

Input:

S =

$$\begin{matrix} 1 & 2 & 2 & 3 & 4 \\ 5 & 3 & 4 & 3 & 1 \\ 4 & 1 & 3 & 2 & 5 \\ 2 & 5 & 3 & 4 & 1 \\ 3 & 4 & 1 & 1 & 5 \end{matrix}$$

i = 1, j = 3, Output: 1 -> 2 -> 4 -> 3

i = 1, j = 2, Output: 1 -> 2

i = 1, j = 4, Output: No ordering => 1 -> 3 -> 2 -> 3 (available in ordering)

i = 3, j = 2, Output: 3 -> 1 -> 2

i = 2, j = 2, Output: 2 -> 3 -> 1 -> 2

i = 1, j = 5, Output: No ordering => 1 -> 4 -> 1 (available in ordering)

i = 2, j = 3, Output: 2 -> 4 -> 3