

Problem 2664: The Knight's Tour

Problem Information

Difficulty: Medium

Acceptance Rate: 72.56%

Paid Only: Yes

Tags: Array, Backtracking, Matrix

Problem Description

Given two positive integers `m` and `n` which are the height and width of a **0-indexed** 2D-array `board`, a pair of positive integers `(r, c)` which is the starting position of the knight on the board.

Your task is to find an order of movements for the knight, in a manner that every cell of the `board` gets visited **exactly** once (the starting cell is considered visited and you **shouldn't** visit it again).

Return `the array board` in which the cells' values show the order of visiting the cell starting from 0 (the initial place of the knight).

Note that a **knight** can **move** from cell `(r1, c1)` to cell `(r2, c2)` if $0 \leq r2 \leq m - 1$ and $0 \leq c2 \leq n - 1$ and $\min(\text{abs}(r1 - r2), \text{abs}(c1 - c2)) = 1$ and $\max(\text{abs}(r1 - r2), \text{abs}(c1 - c2)) = 2$.

Example 1:

Input: `m = 1, n = 1, r = 0, c = 0` **Output:** `[[0]]` **Explanation:** There is only 1 cell and the knight is initially on it so there is only a 0 inside the 1x1 grid.

Example 2:

Input: `m = 3, n = 4, r = 0, c = 0` **Output:** `[[0,3,6,9],[11,8,1,4],[2,5,10,7]]` **Explanation:** By the following order of movements we can visit the entire board.
(0,0)->(1,2)->(2,0)->(0,1)->(1,3)->(2,1)->(0,2)->(2,3)->(1,1)->(0,3)->(2,2)->(1,0)

****Constraints:****

*`1` <= m, n <= 5` *`0` <= r <= m - 1` *`0` <= c <= n - 1` * The inputs will be generated such that there exists at least one possible order of movements with the given condition

Code Snippets

C++:

```
class Solution {
public:
    vector<vector<int>> tourOfKnight(int m, int n, int r, int c) {

    }
};
```

Java:

```
class Solution {
    public int[][] tourOfKnight(int m, int n, int r, int c) {

    }
}
```

Python3:

```
class Solution:
    def tourOfKnight(self, m: int, n: int, r: int, c: int) -> List[List[int]]:
```