

Tên: Nguyễn Hữu Nam
MSSV: 22520917

I. Đề tài: Rat in a maze

- **Input:** số nguyên n , mê cung là mảng 2 chiều ($n \times n$)
- **Output:** mảng bao gồm nhiều chuỗi kí tự là tất cả đường đi để chuột tới vạch đích, trả về mảng rỗng nếu không tìm được đường đi
- **Abstraction:**
 - Tìm đường đi cho chuột trong mê cung ($n \times n$) từ điểm xuất phát $(0,0)$ tới điểm đích $(n-1, n-1)$
- **Pattern recognition:**
 - Để có thể tìm đường đi, chuột sẽ thử hết tất cả trường hợp có thể đi (trên, dưới, trái, phải) và tiếp tục, nếu không tìm được lời giải chuột quay lại và thử lối đi khác
- **Độ phức tạp thuật toán:** $O(4^{n^2})$

II. Thiết kế thuật toán

- **Ý tưởng:** Sử dụng thuật toán backtracking - với mỗi hướng đi, sau mỗi lần thử 4 hướng đi tiếp theo, nếu không có lời giải thì chuột lùi lại 1 bước để thử lối đi khác, thuật toán dừng lại khi chuột đã đi hết tất cả các ô trong mê cung.

Mã giả (pseudocode):

```
direction = [up, left, down, right]
answer = []

def backtracking(matrix, x, y, temp, n):
    if current position is (n - 1, n - 1):
        add temp to answer
        matrix[x][y] = -1                #marked as visited

    for dir in direction:
        if (x + dir == 1 and y + dir == 1) and dir is not out of range:
            temp = temp + dir
            backtracking(matrix, x + dir, y + dir, temp, n)

    matrix[x][y] = 1                    #undo visited
```

Code hoàn chỉnh (python):

```
matrix = #input
n = len(matrix[0])
ans = []
str = ''
def backtracking(matrix, x, y, n, ans, str):
    #1. stop
    if (x == n - 1 and y == n - 1):
        ans.append(str) if str not in ans else None
        return
    matrix[x][y] = -1
    #2. backtrack

    if (x - 1 >= 0 and matrix[x - 1][y] == 1):
        backtracking(matrix, x - 1, y, n, ans, str + 'U') #up
    if (y + 1 < n and matrix[x][y + 1] == 1):
        backtracking(matrix, x, y + 1, n, ans, str + 'R') #right
    if (x + 1 < n and matrix[x + 1][y] == 1):
        backtracking(matrix, x + 1, y, n, ans, str + 'D') #down
    if (y - 1 >= 0 and matrix[x][y - 1] == 1):
        backtracking(matrix, x, y - 1, n, ans, str + 'L') #left

    #3. reverse action
    matrix[x][y] = 1
    return
```