Nhóm 08	CS112.L23.KHCL
Bùi Trí Dũng - 19521386	
Phạm Ngọc Tân - 19520925	
Võ Khánh An - 19520007	

```
import collections

def updateState(state, x, y, x1, y1, x2, y2):
    newState = [[j for j in i] for i in state]
    newState[x][y] = 0
    newState[x + x1][y + y1] = 0
    newState[x + x2][y + y2] = 1
    return newState

def updateAnswer(ans, x, y, type):
    newAns = [i for i in ans]
    newAns.append((x, y, type))
    return newAns


def isLegal(state):
    countOne = 0
    for i in state:
        countOne = countOne + i.count(1)
    if countOne == 1:
        return True
    return False

def solveWithBackTracking(row = 7, column = 7):
    state = [   [2, 2, 1, 1, 1, 2, 2],
                [2, 2, 1, 1, 1, 2, 2],
                [1, 1, 1, 1, 1, 1, 1],
                [1, 1, 1, 0, 1, 1, 1],
                [1, 1, 1, 1, 1, 1, 1],
                [2, 2, 1, 1, 1, 2, 2],
                [2, 2, 1, 1, 1, 2, 2]]


    ans = [(0, 0, 0)]

    visitMap = set()
    stackStt = collections.deque([state])
    stackAns = collections.deque([ans])

    while stackStt:
        top = stackStt.pop()
        ans = stackAns.pop()

        if isLegal(top) == True:
            result = []
            cnt = 0
            for i in ans:
                cnt += 1
                if cnt >= 2:
                    result.append(i)
            return result

        for i in range(0, row):
            for j in range(0, column):
                if i + 2 < row and      top[i][j] == 1 and top[i + 1][j] == 1 and top[i + 2][j]
 == 0:
                    stackStt.append(updateState(top, i, j, 1, 0, 2, 0))
```

```
                    stackAns.append(updateAnswer(ans, i, j, 1))
                if i - 2 >= 0  and      top[i][j] == 1 and top[i - 1][j] == 1 and top[i - 2][j]
 == 0:
                    stackStt.append(updateState(top, i, j,-1, 0,-2, 0))
                    stackAns.append(updateAnswer(ans, i, j, 2))
                if j + 2 < column and   top[i][j] == 1 and top[i][j + 1] == 1 and top[i][j + 2]
 == 0:
                    stackStt.append(updateState(top, i, j, 0, 1, 0, 2))
                    stackAns.append(updateAnswer(ans, i, j, 3))
                if j - 2 >= 0 and       top[i][j] == 1 and top[i][j - 1] == 1 and top[i][j - 2]
 == 0:
                    stackStt.append(updateState(top, i, j, 0,-1, 0,-2))
                    stackAns.append(updateAnswer(ans, i, j, 4))
    return -1


ans = solveWithBackTracking(7, 7)
```