**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

****



**BÁO CÁO THỰC HÀNH**

**MÔN CS106 – TRÍ TUỆ NHÂN TẠO**

**BT1 - DFS/BFS/UCS for Sokoban**

**Giảng viên hướng dẫn: Thầy Lương Ngọc Hoàng**

**Sinh viên thực hiện: Trần Ngọc Thiện**

**MSSV: 21521465**

**TP. HỒ CHÍ MINH, Tháng 3 Năm 2024**

**NHẬN XÉT CỦA GIẢNG VIÊN**

TPHCM, Ngày … Tháng … Năm …

Người nhận xét

(Ký tên)

Mục lục

[Mô tả bài toán 4](#_Toc161788730)

[Cài đặt thuật toán 6](#_Toc161788731)

[Depth first search (DFS) 6](#_Toc161788732)

[Breath first search (BFS) 8](#_Toc161788733)

[Uniform cost search (UCS) 10](#_Toc161788734)

[Cost function 10](#_Toc161788735)

[Cài đặt UCS 10](#_Toc161788736)

[Kết quả 14](#_Toc161788737)

[Depth first search (DFS) 14](#_Toc161788738)

[Breath first search (BFS) 18](#_Toc161788739)

[Uniform cost search (UCS) 19](#_Toc161788740)

[Độ dài đường đi của 3 thuật toán 20](#_Toc161788741)

[Nhận xét 21](#_Toc161788742)

# Mô tả bài toán

Sokoban  là trò chơi dạng câu đố trong đó người chơi phải đẩy một số khối hộp vượt qua chướng ngại vật để đến đích. Trò chơi có dạng bảng ô vuông. Có một số khối hộp được đẩy đến đích (số ô đích đúng bằng số khối hộp). Chỉ có thể đẩy từng khối hộp một, và không thể kéo, cũng như không thể đẩy một dãy hai hay nhiều hộp.

Ảnh có chứa văn bản, ảnh chụp màn hình, máy tính, phần mềm

Mô tả được tạo tự động

Bài toán tìm lời giải cho sokoban được mô hình hóa như sau:

* Không gian trạng thái chứa tất cả tổ hợp vị trí của nhân vật và các khối hộp.
* Mỗi trạng thái là một tổ hợp vị trí của nhân vật và các khối hộp.
* Trạng thái khởi đầu là tổ hợp vị trí của nhân vật và các khối hộp ban đầu của màn chơi.
* Trạng thái kết thúc là tổ hợp vị trí của nhân vật và các khối hộp mà trong đó vị trí mỗi hộp trùng với vị trí một ô đích.
* Successor function: cập nhật vị trí nhân vật và các khối hộp
  + Hành động (action): di chuyển nhân vật về 1 trong 4 hướng lên (up, u), xuống (down, d), trái (left, l), phải (right, r) với cự li 1 ô; nếu tồn tại khối hộp tại đích thì di chuyển khối hộp đó 1 ô theo cùng hướng di chuyển của nhân vật.
  + Chi phí (cost) (với uniform cost search): chi phí đường đi là số bước đi không dịch chuyển hộp.
* Các hành động hợp lệ khi:
  + Với hành động không dịch chuyển hộp: vị trí đích đến của nhân vật không có chướng ngại (tường)
  + Với hành động có dịch chuyển hộp: vị trí đích đến của hộp không có chướng ngại (tường, hộp khác)
* Output bài toán: lời giải cho màn chơi (tổ hợp các hành động theo thứ tụ để biến đổi từ trạng thái khởi đầu đến trạng thái kết thúc).

# Cài đặt thuật toán

## Depth first search (DFS)

def depthFirstSearch(gameState):

    """Implement depthFirstSearch approach"""

    # Input: gameState: trạng thái khởi đầu của màn chơi

    # Output: lời giải của màn chơi

    beginBox = PosOfBoxes(gameState)    # vị trí khởi đầu của box

    beginPlayer = PosOfPlayer(gameState)    # vị trí khởi đầu của player

    # trạng thái bài toán = ( (vị trí agent / player), (vị trí các box) )

    startState = (beginPlayer, beginBox)    # trạng thái khởi đầu

    # hàng đợi chứa các node sắp đuọc bung ra

    # mỗi node chứa các trạng thái theo thứ tự tạo thành một lời giải / solution

    # mỗi node là một list chứa các list con, mỗi list con chứa 1 trạng thái

    frontier = collections.deque([[startState]])

    exploredSet = set() # set chứa trạng thái đã duyệt qua

    actions = [[0]] # list chứa list các action của các solution

    temp = []   # return list

    while frontier:

        # lấy node bên phải frontier / node mới nhất trong frontier / node sâu nhất

        node = frontier.pop()

        # lấy list action ứng với node vừa lấy

        node\_action = actions.pop()

        # tt đang duyệt là end state

        if isEndState(node[-1][-1]):

            temp += node\_action[1:] # list action -> temp

            break

        # end\_if

        # tt đang duyệt chưa đc duyệt qua

        if node[-1] not in exploredSet:

            exploredSet.add(node[-1])

            # duyệt qua từng legal action của tt hiện tại

            for action in legalActions(node[-1][0], node[-1][1]):

                # tt mới sau khi thực hiện action

                newPosPlayer, newPosBox = updateState(node[-1][0], node[-1][1], action)

                # tt mới có thể không dẫn tới solution

                if isFailed(newPosBox):

                    continue    # bỏ qua

                # end\_if

                # thêm tt mới vào node rồi thêm node vào bên phải frontier

                frontier.append(node + [(newPosPlayer, newPosBox)])

                # thêm action mới thực hiện vào list action rồi thêm vào list list actions

                actions.append(node\_action + [action[-1]])

            # end\_for

        # end\_if

    # end\_while

    return temp

# end\_def

## Breath first search (BFS)

def breadthFirstSearch(gameState):

    """Implement breadthFirstSearch approach"""

    # Input: gameState: trạng thái khởi đầu của màn chơi

    # Output: lời giải của màn chơi

    beginBox = PosOfBoxes(gameState)    # vị trí khởi đầu của box

    beginPlayer = PosOfPlayer(gameState)    # vị trí khởi đầu của player

    # trạng thái bài toán = ( (vị trí agent / player), (vị trí các box) )

    startState = (beginPlayer, beginBox)    # trạng thái khởi đầu

    # hàng đợi chứa các node sắp đuọc bung ra

    # mỗi node chứa các trạng thái theo thứ tự tạo thành một lời giải / solution

    # mỗi node là một list chứa các list con, mỗi list con chứa 1 trạng thái

    frontier = collections.deque([[startState]])

    exploredSet = set() # set chứa trạng thái đã duyệt qua

    actions = collections.deque([[0]]) # list chứa list các action của các solution

    temp = []   # return list

    ### CODING FROM HERE ###

    while frontier:

        # lấy node bên trái frontier / node cũ nhất trong frontier / node tiếp theo trên level hiện tại

        node = frontier.popleft()

        # lấy list action ứng với node vừa lấy

        node\_action = actions.popleft()

        # tt đang duyệt là end state

        if isEndState(node[-1][-1]):

            temp += node\_action[1:] # list action -> temp

            break

        # end\_if

        # tt đang duyệt chưa đc duyệt qua

        if node[-1] not in exploredSet:

            exploredSet.add(node[-1])

            # duyệt qua từng legal action của tt hiện tại

            for action in legalActions(node[-1][0], node[-1][1]):

                # tt mới sau khi thực hiện action

                newPosPlayer, newPosBox = updateState(node[-1][0], node[-1][1], action)

                # tt mới có thể không dẫn tới solution

                if isFailed(newPosBox):

                    continue    # bỏ qua

                # end\_if

                # thêm tt mới vào node rồi thêm node vào bên phải frontier

                frontier.append(node + [(newPosPlayer, newPosBox)])

                # thêm action mới thực hiện vào list action rồi thêm vào list list actions

                actions.append(node\_action + [action[-1]])

            # end\_for

        # end\_if

    # end\_while

    return temp

# end\_def

## Uniform cost search (UCS)

### Cost function

def cost(actions):

    """A cost function"""

    # cost = số action không di chuyển box

    return len([x for x in actions if x.islower()])

### Cài đặt UCS

def uniformCostSearch(gameState):

    """Implement uniformCostSearch approach"""

    # Input: gameState: trạng thái khởi đầu của màn chơi

    # Output: lời giải của màn chơi

    beginBox = PosOfBoxes(gameState)    # vị trí khởi đầu của box

    beginPlayer = PosOfPlayer(gameState)    # vị trí khởi đầu của player

    # trạng thái bài toán = ( (vị trí agent / player), (vị trí các box) )

    startState = (beginPlayer, beginBox)    # trạng thái khởi đầu

    # hàng đợi chứa các node sắp đuọc bung ra

    # mỗi node chứa các trạng thái theo thứ tự tạo thành một lời giải / solution

    # mỗi node là một list chứa các list con, mỗi list con chứa 1 trạng thái

    frontier = PriorityQueue()

    frontier.push([startState], 0)

    exploredSet = set() # set chứa trạng thái đã duyệt qua

    # list chứa list các action của các solution

    actions = PriorityQueue()

    actions.push([0], 0)

    temp = []   # return list

    ### CODING FROM HERE ###

    while (frontier.isEmpty == False):

        # lấy node có cost thấp nhất frontier

        node = frontier.pop()

        # lấy list action ứng với node vừa lấy

        node\_action = actions.pop()

        # tt đang duyệt là end state

        if isEndState(node[-1][-1]):

            temp += node\_action[1:] # list action -> temp

            break

        # end\_if

        # tt đang duyệt chưa đc duyệt qua

        if node[-1] not in exploredSet:

            exploredSet.add(node[-1])

            # duyệt qua từng legal action của tt hiện tại

            for action in legalActions(node[-1][0], node[-1][1]):

                # tt mới sau khi thực hiện action

                newPosPlayer, newPosBox = updateState(node[-1][0], node[-1][1], action)

                # tt mới có thể không dẫn tới solution

                if isFailed(newPosBox):

                    continue    # bỏ qua

                # end\_if

                new\_node = node + [(newPosPlayer, newPosBox)]   # thêm tt mới vào node

                new\_action = node\_action + [action[-1]] # thêm action mới thực hiện vào list action

                # tính cost

                new\_cost = cost(new\_action[1:])

                # thêm node vào bên phải frontier

                frontier.push(new\_node, new\_cost)

                # thêm vào list list actions

                actions.push(new\_action, new\_cost)

            # end\_for

        # end\_if

    # end\_while

    return temp

# end\_def

# Kết quả

## Depth first search (DFS)

|  |  |  |  |
| --- | --- | --- | --- |
| Level | Thời gian (giây) | Lời giải | Số bước đi |
| 1 | 0.08 | ['r', 'r', 'U', 'r', 'd', 'l', 'l', 'l', 'u', 'R', 'R', 'l', 'l', 'd', 'r', 'r', 'r', 'U', 'l', 'l', 'l', 'u', 'r', 'u', 'r', 'r', 'D', 'L', 'r', 'u', 'l', 'l', 'l', 'd', 'R', 'l', 'd', 'r', 'r', 'U', 'r', 'u', 'L', 'r', 'd', 'l', 'l', 'l', 'd', 'r', 'r', 'd', 'r', 'U', 'l', 'l', 'l', 'u', 'r', 'r', 'u', 'r', 'D', 'l', 'u', 'L', 'r', 'r', 'd', 'l', 'l', 'l', 'd', 'r', 'r', 'd', 'r', 'U', 'U'] | 79 |
| 2 | 0 | ['r', 'r', 'r', 'U', 'l', 'l', 'd', 'l', 'U', 'r', 'r', 'r', 'U', 'l', 'l', 'd', 'r', 'r', 'd', 'l', 'l', 'l', 'u', 'U'] | 24 |
| 3 | 0.3 | ['r', 'r', 'd', 'l', 'd', 'r', 'd', 'l', 'l', 'D', 'R', 'l', 'l', 'd', 'l', 'U', 'r', 'r', 'r', 'd', 'r', 'r', 'u', 'L', 'r', 'd', 'l', 'l', 'L', 'r', 'r', 'r', 'u', 'l', 'u', 'l', 'l', 'd', 'R', 'l', 'l', 'l', 'd', 'R', 'l', 'u', 'r', 'r', 'r', 'd', 'r', 'r', 'u', 'L', 'r', 'd', 'l', 'l', 'U', 'r', 'r', 'd', 'l', 'l', 'L', 'r', 'r', 'r', 'u', 'l', 'l', 'l', 'l', 'l', 'd', 'R', 'R', 'R', 'l', 'l', 'l', 'u', 'r', 'r', 'r', 'r', 'r', 'd', 'L', 'r', 'u', 'l', 'l', 'l', 'l', 'l', 'U', 'd', 'r', 'r', 'r', 'r', 'r', 'd', 'l', 'L', 'r', 'r', 'u', 'l', 'l', 'l', 'l', 'l', 'd', 'r', 'R', 'R', 'l', 'l', 'l', 'u', 'r', 'r', 'r', 'r', 'r', 'd', 'L', 'r', 'u', 'l', 'l', 'l', 'u', 'R', 'l', 'd', 'r', 'r', 'r', 'd', 'l', 'L', 'r', 'r', 'u', 'l', 'l', 'l', 'l', 'l', 'd', 'r', 'R', 'R', 'l', 'l', 'l', 'u', 'r', 'r', 'r', 'r', 'r', 'd', 'L', 'r', 'u', 'l', 'l', 'l', 'u', 'r', 'u', 'r', 'u', 'l', 'l', 'u', 'L', 'r', 'r', 'r', 'd', 'l', 'd', 'd', 'l', 'd', 'r', 'r', 'r', 'd', 'l', 'L', 'r', 'r', 'u', 'l', 'l', 'l', 'l', 'l', 'd', 'r', 'R', 'R', 'l', 'l', 'l', 'u', 'r', 'r', 'r', 'r', 'r', 'd', 'L', 'r', 'u', 'l', 'U', 'l', 'l', 'd', 'r', 'r', 'r', 'd', 'l', 'L', 'r', 'r', 'u', 'l', 'l', 'l', 'l', 'l', 'd', 'r', 'R', 'R', 'l', 'l', 'l', 'u', 'r', 'r', 'r', 'r', 'r', 'd', 'L', 'r', 'u', 'l', 'l', 'l', 'u', 'r', 'r', 'U', 'l', 'd', 'r', 'd', 'r', 'd', 'l', 'L', 'r', 'r', 'u', 'l', 'l', 'l', 'l', 'l', 'd', 'r', 'R', 'R', 'l', 'l', 'l', 'u', 'r', 'r', 'r', 'r', 'r', 'd', 'L', 'r', 'u', 'l', 'l', 'l', 'u', 'r', 'r', 'u', 'l', 'u', 'l', 'u', 'r', 'r', 'D', 'D', 'l', 'd', 'l', 'd', 'r', 'r', 'r', 'd', 'l', 'L', 'r', 'r', 'u', 'l', 'l', 'l', 'u', 'r', 'u', 'r', 'D', 'l', 'l', 'd', 'r', 'd', 'r', 'r', 'u', 'L', 'r', 'd', 'l', 'l', 'L', 'r', 'r', 'r', 'u', 'l', 'u', 'l', 'l', 'd', 'R', 'l', 'l', 'l', 'd', 'R', 'l', 'u', 'r', 'r', 'r', 'd', 'L', 'r', 'r', 'r', 'u', 'L', 'r', 'd', 'l', 'l', 'l', 'u', 'l', 'l', 'd', 'R', 'R', 'R', 'l', 'l', 'l', 'u', 'r', 'r', 'u', 'r', 'r', 'd', 'r', 'd', 'L', 'r', 'u', 'l', 'L'] | 403 |
| 4 | 0 | ['r', 'r', 'r', 'd', 'l', 'l', 'l', 'l', 'L', 'r', 'r', 'r', 'r', 'r', 'd', 'l', 'l', 'l', 'l', 'l', 'l', 'l', 'u', 'R', 'R', 'R', 'R'] | 27 |
| 5 |  | Không tìm được trên máy của em | 0 |
| 6 | 0.02 | ['r', 'r', 'd', 'l', 'l', 'l', 'l', 'u', 'R', 'R', 'l', 'l', 'd', 'r', 'r', 'r', 'r', 'u', 'L', 'r', 'd', 'l', 'l', 'U', 'U', 'U', 'd', 'd', 'r', 'r', 'd', 'l', 'l', 'l', 'l', 'u', 'u', 'u', 'u', 'r', 'R', 'l', 'l', 'd', 'd', 'd', 'r', 'r', 'r', 'r', 'u', 'u', 'u', 'L', 'L'] | 55 |
| 7 | 0.74 | ['r', 'd', 'l', 'l', 'l', 'l', 'u', 'r', 'R', 'l', 'l', 'd', 'r', 'r', 'r', 'r', 'u', 'u', 'l', 'L', 'r', 'r', 'd', 'L', 'r', 'd', 'l', 'l', 'l', 'l', 'u', 'r', 'R', 'l', 'l', 'd', 'r', 'r', 'r', 'r', 'u', 'u', 'l', 'l', 'L', 'r', 'r', 'r', 'd', 'L', 'r', 'd', 'l', 'l', 'l', 'l', 'u', 'r', 'R', 'l', 'l', 'd', 'r', 'r', 'r', 'r', 'u', 'u', 'l', 'l', 'u', 'u', 'l', 'l', 'd', 'D', 'r', 'r', 'r', 'r', 'd', 'L', 'r', 'd', 'l', 'l', 'l', 'l', 'U', 'r', 'd', 'r', 'r', 'r', 'u', 'l', 'L', 'r', 'r', 'd', 'l', 'l', 'l', 'l', 'u', 'U', 'r', 'r', 'r', 'r', 'd', 'l', 'l', 'L', 'r', 'r', 'r', 'u', 'l', 'l', 'l', 'l', 'U', 'd', 'r', 'r', 'r', 'r', 'd', 'l', 'l', 'l', 'd', 'l', 'U', 'r', 'r', 'r', 'r', 'u', 'l', 'l', 'u', 'u', 'l', 'u', 'l', 'l', 'd', 'R', 'R', 'l', 'l', 'u', 'r', 'r', 'r', 'r', 'r', 'd', 'd', 'd', 'l', 'l', 'l', 'd', 'r', 'r', 'r', 'd', 'l', 'l', 'l', 'l', 'u', 'U', 'r', 'r', 'r', 'r', 'u', 'u', 'u', 'l', 'l', 'l', 'l', 'l', 'd', 'r', 'D', 'D', 'r', 'r', 'r', 'r', 'd', 'l', 'l', 'l', 'd', 'l', 'U', 'r', 'r', 'r', 'r', 'u', 'l', 'l', 'u', 'U', 'l', 'l', 'l', 'u', 'r', 'r', 'R', 'l', 'l', 'l', 'd', 'r', 'r', 'r', 'd', 'd', 'r', 'r', 'd', 'l', 'l', 'l', 'l', 'U', 'r', 'r', 'r', 'r', 'u', 'u', 'u', 'L', 'r', 'd', 'd', 'd', 'l', 'l', 'l', 'l', 'U', 'd', 'r', 'r', 'u', 'u', 'l', 'u', 'R', 'l', 'l', 'l', 'd', 'R', 'R', 'l', 'l', 'u', 'r', 'r', 'r', 'D', 'l', 'l', 'd', 'd', 'r', 'r', 'r', 'r', 'u', 'u', 'u', 'L', 'r', 'd', 'd', 'd', 'l', 'l', 'l', 'l', 'u', 'u', 'r', 'r', 'D', 'D', 'r', 'r', 'd', 'l', 'L', 'r', 'r', 'd', 'l', 'l', 'l', 'l', 'u', 'R', 'R', 'l', 'l', 'd', 'r', 'r', 'r', 'r', 'u', 'u', 'l', 'l', 'l', 'l', 'u', 'u', 'r', 'u', 'R', 'l', 'l', 'l', 'd', 'r', 'r', 'r', 'd', 'd', 'r', 'r', 'd', 'L', 'r', 'd', 'l', 'l', 'l', 'l', 'u', 'r', 'R', 'l', 'l', 'd', 'r', 'r', 'r', 'r', 'u', 'u', 'u', 'u', 'u', 'L', 'r', 'd', 'd', 'd', 'l', 'l', 'l', 'l', 'd', 'r', 'r', 'd', 'r', 'U', 'r', 'd', 'l', 'l', 'l', 'l', 'u', 'r', 'r', 'u', 'l', 'l', 'u', 'u', 'r', 'u', 'R', 'l', 'l', 'l', 'd', 'r', 'r', 'r', 'd', 'd', 'l', 'l', 'd', 'r', 'r', 'r', 'r', 'u', 'L', 'r', 'd', 'l', 'l', 'l', 'l', 'u', 'r', 'R', 'l', 'l', 'd', 'r', 'r', 'r', 'r', 'u', 'u', 'u', 'u', 'L', 'r', 'd', 'd', 'd', 'L', 'r', 'd', 'l', 'l', 'l', 'l', 'u', 'u', 'u', 'r', 'u', 'R', 'l', 'l', 'l', 'd', 'r', 'r', 'r', 'd', 'D', 'r', 'r', 'u', 'u', 'u', 'L', 'r', 'd', 'd', 'd', 'l', 'l', 'l', 'l', 'u', 'u', 'r', 'u', 'R', 'l', 'l', 'l', 'd', 'r', 'r', 'r', 'd', 'd', 'r', 'r', 'd', 'l', 'L', 'r', 'r', 'd', 'l', 'l', 'l', 'l', 'u', 'u', 'r', 'r', 'r', 'r', 'u', 'u', 'u', 'L', 'r', 'd', 'd', 'd', 'l', 'l', 'l', 'l', 'd', 'd', 'r', 'r', 'r', 'r', 'u', 'l', 'l', 'L', 'r', 'r', 'r', 'd', 'l', 'l', 'l', 'l', 'U', 'r', 'r', 'r', 'r', 'u', 'l', 'l', 'u', 'u', 'l', 'l', 'd', 'D', 'r', 'r', 'r', 'r', 'u', 'u', 'u', 'l', 'L', 'r', 'r', 'd', 'd', 'd', 'l', 'l', 'l', 'l', 'u', 'u', 'l', 'u', 'r', 'R', 'R', 'l', 'l', 'l', 'd', 'r', 'r', 'r', 'd', 'd', 'r', 'r', 'd', 'l', 'l', 'l', 'd', 'l', 'U', 'r', 'r', 'r', 'r', 'u', 'u', 'u', 'u', 'L', 'r', 'd', 'd', 'd', 'l', 'l', 'l', 'd', 'r', 'r', 'r', 'd', 'l', 'l', 'l', 'l', 'u', 'U', 'r', 'r', 'r', 'r', 'u', 'u', 'u', 'l', 'L', 'r', 'r', 'd', 'd', 'd', 'l', 'l', 'u', 'u', 'l', 'l', 'l', 'u', 'r', 'R', 'R', 'l', 'l', 'l', 'd', 'r', 'r', 'r', 'd', 'd', 'r', 'r', 'd', 'l', 'l', 'l', 'l', 'u', 'U', 'd', 'r', 'r', 'r', 'r', 'u', 'u', 'u', 'L', 'r', 'd', 'd', 'd', 'l', 'l', 'l', 'l', 'u', 'U', 'r', 'u', 'R', 'l', 'd', 'r', 'd', 'd', 'r', 'r', 'd', 'l', 'l', 'l', 'l', 'u', 'u', 'u', 'l', 'u', 'R'] | 707 |
| 8 | 0.1 | ['l', 'l', 'D', 'l', 'u', 'r', 'r', 'r', 'd', 'L', 'r', 'u', 'l', 'l', 'l', 'd', 'R', 'D', 'D', 'D', 'D', 'D', 'l', 'l', 'l', 'd', 'd', 'r', 'r', 'd', 'r', 'r', 'u', 'u', 'L', 'r', 'd', 'd', 'l', 'l', 'u', 'l', 'l', 'u', 'u', 'r', 'r', 'r', 'u', 'u', 'u', 'u', 'u', 'l', 'u', 'r', 'r', 'r', 'd', 'L', 'r', 'u', 'l', 'l', 'l', 'd', 'R', 'd', 'd', 'd', 'd', 'd', 'l', 'l', 'l', 'd', 'd', 'r', 'r', 'U', 'r', 'u', 'L', 'r', 'u', 'u', 'u', 'u', 'u', 'l', 'u', 'r', 'r', 'r', 'd', 'L', 'r', 'u', 'l', 'l', 'D', 'D', 'D', 'D', 'D', 'D', 'l', 'd', 'd', 'l', 'l', 'u', 'u', 'R', 'R', 'l', 'l', 'd', 'd', 'r', 'r', 'd', 'r', 'r', 'u', 'u', 'L', 'r', 'd', 'd', 'l', 'l', 'u', 'l', 'l', 'u', 'u', 'r', 'r', 'D', 'r', 'U', 'l', 'd', 'r', 'r', 'd', 'd', 'l', 'l', 'U', 'l', 'l', 'u', 'u', 'r', 'r', 'r', 'U', 'd', 'l', 'l', 'l', 'd', 'd', 'r', 'r', 'U', 'r', 'u', 'L', 'r', 'u', 'U', 'd', 'd', 'l', 'd', 'r', 'r', 'd', 'd', 'l', 'l', 'u', 'l', 'l', 'u', 'u', 'R', 'l', 'd', 'd', 'r', 'r', 'd', 'r', 'r', 'u', 'u', 'l', 'u', 'L', 'r', 'u', 'u', 'U', 'd', 'd', 'd', 'l', 'd', 'r', 'r', 'd', 'd', 'l', 'l', 'u', 'l', 'l', 'u', 'u', 'R', 'l', 'd', 'd', 'r', 'r', 'd', 'r', 'r', 'u', 'u', 'l', 'u', 'L', 'r', 'u', 'u', 'u', 'U', 'd', 'd', 'd', 'd', 'l', 'd', 'r', 'r', 'd', 'd', 'l', 'l', 'u', 'l', 'l', 'u', 'u', 'R', 'l', 'd', 'd', 'r', 'r', 'd', 'r', 'r', 'u', 'u', 'l', 'u', 'L', 'r', 'u', 'u', 'u', 'u', 'U', 'l', 'u', 'R', 'l', 'd', 'r', 'd', 'd', 'd', 'd', 'd', 'l', 'd', 'r', 'r', 'd', 'd', 'l', 'l', 'u', 'l', 'l', 'u', 'u', 'R', 'R', 'l', 'l', 'd', 'd', 'r', 'r', 'd', 'r', 'r', 'u', 'u', 'l', 'U', 'U', 'U', 'U', 'U', 'U'] | 323 |
| 9 | 0.37 | ['d', 'l', 'l', 'l', 'U', 'l', 'u', 'R', 'R', 'l', 'l', 'd', 'r', 'd', 'r', 'r', 'r', 'u', 'u', 'L', 'L', 'r', 'r', 'd', 'd', 'l', 'l', 'U', 'l', 'l', 'u', 'u', 'r', 'D', 'R', 'l', 'l', 'd', 'R', 'd', 'r', 'r', 'r', 'u', 'u', 'L', 'L', 'r', 'r', 'd', 'd', 'l', 'l', 'l', 'u', 'l', 'u', 'u', 'r', 'D', 'r', 'r', 'r', 'd', 'd', 'l', 'U', 'r', 'd', 'l', 'l', 'l', 'U', 'R'] | 74 |
| 10 | 0.02 | ['d', 'l', 'U', 'r', 'r', 'r', 'd', 'L', 'r', 'u', 'l', 'l', 'l', 'd', 'd', 'r', 'U', 'l', 'u', 'R', 'u', 'u', 'l', 'D', 'r', 'd', 'd', 'r', 'r', 'u', 'L', 'r', 'd', 'l', 'l', 'U', 'U'] | 37 |
| 11 | 0.02 | ['r', 'D', 'l', 'd', 'd', 'r', 'r', 'r', 'u', 'u', 'L', 'L', 'r', 'r', 'd', 'd', 'l', 'l', 'U', 'l', 'd', 'r', 'r', 'r', 'u', 'u', 'l', 'u', 'l', 'l', 'D', 'R', 'd', 'd', 'l', 'U'] | 36 |
| 12 | 0.2 | ['r', 'r', 'D', 'r', 'r', 'd', 'l', 'd', 'l', 'l', 'l', 'u', 'u', 'R', 'R', 'l', 'l', 'd', 'd', 'r', 'r', 'U', 'r', 'r', 'd', 'l', 'l', 'l', 'l', 'u', 'u', 'r', 'u', 'r', 'r', 'D', 'r', 'd', 'd', 'l', 'l', 'l', 'l', 'u', 'u', 'r', 'R', 'l', 'l', 'd', 'd', 'r', 'r', 'r', 'r', 'u', 'L', 'r', 'd', 'l', 'l', 'l', 'l', 'u', 'u', 'r', 'r', 'u', 'r', 'D', 'r', 'd', 'd', 'l', 'l', 'U', 'R', 'l', 'd', 'l', 'l', 'u', 'u', 'r', 'u', 'r', 'r', 'd', 'L', 'r', 'd', 'l', 'd', 'l', 'l', 'u', 'u', 'R', 'l', 'd', 'd', 'r', 'r', 'r', 'u', 'u', 'u', 'l', 'D'] | 109 |
| 13 | 0.26 | ['r', 'r', 'd', 'l', 'l', 'd', 'd', 'r', 'r', 'r', 'r', 'u', 'u', 'L', 'r', 'd', 'l', 'l', 'd', 'l', 'l', 'u', 'u', 'r', 'u', 'r', 'D', 'r', 'r', 'd', 'l', 'd', 'l', 'l', 'l', 'u', 'u', 'r', 'u', 'r', 'u', 'r', 'D', 'l', 'l', 'l', 'd', 'd', 'd', 'r', 'r', 'r', 'r', 'u', 'u', 'L', 'L', 'r', 'r', 'd', 'l', 'd', 'l', 'U', 'U', 'r', 'r', 'd', 'l', 'l', 'd', 'l', 'l', 'u', 'u', 'R', 'R', 'l', 'l', 'd', 'd', 'r', 'r', 'r', 'r', 'u', 'u', 'L', 'r', 'd', 'l', 'l', 'd', 'l', 'l', 'u', 'u', 'u', 'r', 'R', 'l', 'l', 'd', 'd', 'd', 'r', 'r', 'r', 'r', 'u', 'l', 'u', 'L', 'r', 'r', 'd', 'l', 'l', 'd', 'l', 'l', 'u', 'u', 'u', 'r', 'r', 'u', 'r', 'D', 'l', 'l', 'l', 'd', 'R', 'l', 'u', 'r', 'r', 'r', 'D', 'r', 'd', 'd', 'l', 'l', 'l', 'l', 'u', 'u', 'r', 'R', 'l', 'l', 'd', 'd', 'r', 'r', 'r', 'r', 'u', 'L', 'r', 'd', 'l', 'l', 'l', 'l', 'u', 'u', 'r', 'r', 'u', 'r', 'D', 'r', 'd', 'd', 'l', 'l', 'U', 'U', 'r', 'r', 'd', 'L'] | 185 |
| 14 | 5.29 | ['d', 'l', 'l', 'l', 'l', 'l', 'u', 'r', 'u', 'r', 'r', 'r', 'D', 'r', 'd', 'L', 'r', 'u', 'l', 'u', 'l', 'l', 'l', 'l', 'd', 'r', 'd', 'r', 'R', 'l', 'l', 'l', 'u', 'r', 'u', 'r', 'r', 'D', 'r', 'r', 'd', 'd', 'l', 'U', 'r', 'u', 'L', 'r', 'd', 'l', 'L', 'L', 'r', 'r', 'r', 'u', 'l', 'u', 'l', 'D', 'r', 'r', 'u', 'l', 'l', 'l', 'l', 'l', 'd', 'r', 'R', 'R', 'l', 'l', 'l', 'u', 'r', 'r', 'r', 'r', 'r', 'd', 'L', 'L', 'L', 'd', 'R', 'l', 'u', 'r', 'r', 'r', 'd', 'd', 'l', 'U', 'r', 'u', 'L', 'r', 'd', 'l', 'l', 'l', 'u', 'R', 'l', 'u', 'l', 'l', 'd', 'R', 'R', 'l', 'l', 'd', 'R', 'R', 'R', 'l', 'l', 'l', 'u', 'r', 'r', 'u', 'r', 'r', 'r', 'd', 'd', 'L', 'r', 'u', 'u', 'l', 'D', 'r', 'd', 'd', 'l', 'U', 'r', 'u', 'u', 'l', 'l', 'l', 'l', 'l', 'd', 'r', 'r', 'd', 'R', 'l', 'l', 'l', 'u', 'r', 'r', 'u', 'r', 'r', 'r', 'd', 'd', 'L', 'L', 'r', 'r', 'u', 'u', 'l', 'D', 'r', 'd', 'd', 'l', 'U', 'l', 'L', 'r', 'r', 'r', 'u', 'u', 'l', 'l', 'D', 'l', 'd', 'R', 'l', 'u', 'r', 'u', 'r', 'r', 'd', 'L', 'r', 'd', 'L', 'r', 'u', 'l', 'u', 'l', 'l', 'l', 'l', 'd', 'r', 'r', 'd', 'R', 'l', 'u', 'R', 'l', 'l', 'l', 'd', 'R', 'R', 'l', 'l', 'u', 'r', 'r', 'r', 'u', 'r', 'r', 'd', 'L', 'r', 'd', 'd', 'l', 'U', 'r', 'u', 'u', 'l', 'D', 'r', 'u', 'l', 'l', 'l', 'l', 'l', 'd', 'r', 'r', 'R', 'l', 'l', 'l', 'u', 'r', 'r', 'r', 'r', 'r', 'd', 'L', 'L', 'r', 'r', 'd', 'd', 'l', 'U', 'r', 'u', 'u', 'l', 'l', 'l', 'l', 'l', 'd', 'r', 'R', 'l', 'l', 'u', 'r', 'r', 'r', 'r', 'r', 'd', 'd', 'l', 'L', 'L', 'r', 'r', 'r', 'u', 'u', 'l', 'D', 'r', 'd', 'L', 'r', 'u', 'l', 'u', 'l', 'l', 'l', 'l', 'd', 'r', 'r', 'd', 'R', 'l', 'u', 'l', 'l', 'd', 'R', 'R', 'l', 'l', 'u', 'r', 'r', 'u', 'r', 'r', 'r', 'd', 'l', 'L', 'L', 'r', 'r', 'r', 'd', 'd', 'l', 'U', 'r', 'u', 'L', 'r', 'd', 'l', 'L', 'r', 'r', 'u', 'l', 'L', 'r', 'r', 'd', 'l', 'l', 'L', 'r', 'r', 'r', 'u', 'l', 'l', 'u', 'l', 'l', 'l', 'd', 'd', 'R', 'R', 'R', 'l', 'l', 'l', 'u', 'u', 'r', 'r', 'r', 'r', 'r', 'd', 'd', 'L', 'r', 'u', 'l', 'l', 'u', 'l', 'l', 'D', 'R', 'R', 'l', 'd', 'R', 'l', 'u', 'r', 'u', 'l', 'l', 'l', 'd', 'd', 'R', 'l', 'u', 'r', 'r', 'r', 'd', 'L', 'r', 'u', 'l', 'l', 'l', 'u', 'r', 'r', 'r', 'r', 'r', 'd', 'L', 'r', 'd', 'd', 'l', 'U', 'r', 'u', 'u', 'l', 'D', 'L', 'r', 'r', 'd', 'd', 'l', 'U', 'r', 'u', 'u', 'l', 'l', 'l', 'l', 'l', 'd', 'd', 'R', 'R', 'R', 'l', 'l', 'l', 'u', 'r', 'u', 'r', 'r', 'r', 'r', 'd', 'L', 'r', 'u', 'l', 'l', 'l', 'D', 'R', 'l', 'l', 'l', 'u', 'r', 'r', 'r', 'r', 'r', 'd', 'L', 'r', 'd', 'd', 'l', 'U', 'r', 'u', 'u', 'l', 'D', 'r', 'u', 'l', 'l', 'l', 'l', 'l', 'd', 'r', 'r', 'R', 'l', 'l', 'l', 'u', 'r', 'r', 'r', 'r', 'r', 'd', 'L', 'L', 'r', 'r', 'd', 'd', 'l', 'U', 'r', 'u', 'u', 'l', 'l', 'd', 'L', 'r', 'd', 'r', 'r', 'u', 'L', 'r', 'd', 'l', 'l', 'L', 'r', 'r', 'r', 'u', 'l', 'u', 'l', 'l', 'l', 'l', 'd', 'R', 'l', 'd', 'R', 'R', 'R', 'l', 'l', 'l', 'u', 'r', 'u', 'r', 'r', 'r', 'r', 'd', 'd', 'd', 'l', 'U', 'r', 'u', 'u', 'l', 'l', 'D', 'L', 'r', 'u', 'r', 'r', 'd', 'd', 'l', 'L', 'r', 'r', 'u', 'L', 'r', 'u', 'l', 'l', 'l', 'l', 'l', 'd', 'd', 'r', 'R', 'R', 'l', 'l', 'l', 'u', 'R', 'l', 'u', 'r', 'r', 'D', 'l', 'l', 'u', 'r', 'r', 'r', 'r', 'r', 'd', 'l', 'L', 'L', 'r', 'r', 'r', 'd', 'd', 'l', 'U', 'r', 'u', 'u', 'l', 'l', 'l', 'l', 'l', 'd', 'R', 'l', 'd', 'r', 'R', 'R', 'l', 'l', 'l', 'u', 'r', 'u', 'r', 'r', 'r', 'r', 'd', 'd', 'L', 'L', 'r', 'r', 'u', 'u', 'l', 'l', 'd', 'L', 'r', 'd', 'r', 'r', 'u', 'u', 'l', 'l', 'l', 'l', 'l', 'd', 'R', 'l', 'u', 'r', 'r', 'r', 'r', 'r', 'd', 'd', 'l', 'l', 'L', 'r', 'r', 'r', 'u', 'u', 'l', 'l', 'd', 'L', 'r', 'd', 'r', 'r', 'u', 'u', 'l', 'l', 'l', 'l', 'l', 'd', 'R', 'l', 'u', 'r', 'r', 'r', 'r', 'D', 'r', 'u', 'l', 'l', 'l', 'l', 'l', 'd', 'd', 'R', 'l', 'u', 'r', 'u', 'r', 'r', 'r', 'r', 'd', 'l', 'l', 'd', 'L', 'r', 'u', 'r', 'r', 'd', 'd', 'l', 'U', 'r', 'u', 'u', 'l', 'l', 'd', 'L', 'r', 'd', 'r', 'r', 'u', 'u', 'l', 'l', 'l', 'l', 'l', 'd', 'd', 'R', 'R', 'R', 'l', 'l', 'l', 'u', 'R', 'l', 'd', 'r', 'r', 'r', 'u', 'L', 'r', 'd', 'l', 'l', 'l', 'u', 'u', 'r', 'r', 'r', 'r', 'r', 'd', 'L', 'r', 'd', 'L', 'r', 'u', 'l', 'u', 'l', 'l', 'd', 'R', 'l', 'd', 'R', 'l', 'l', 'l', 'u', 'u', 'r', 'r', 'r', 'r', 'r', 'd', 'L', 'r', 'd', 'd', 'l', 'U', 'r', 'u', 'u', 'l', 'D', 'L', 'r', 'r', 'd', 'd', 'l', 'U', 'r', 'u', 'L'] | 865 |
| 15 | 0.22 | ['r', 'r', 'd', 'l', 'd', 'L', 'r', 'u', 'r', 'u', 'l', 'l', 'D', 'l', 'D', 'l', 'u', 'r', 'r', 'r', 'd', 'L', 'r', 'u', 'r', 'u', 'l', 'l', 'd', 'l', 'l', 'd', 'R', 'D', 'D', 'r', 'd', 'L', 'r', 'u', 'l', 'u', 'u', 'l', 'u', 'r', 'r', 'r', 'd', 'L', 'r', 'u', 'r', 'u', 'l', 'l', 'd', 'l', 'l', 'd', 'R', 'd', 'd', 'r', 'd', 'l', 'L', 'r', 'r', 'u', 'l', 'u', 'u', 'l', 'u', 'r', 'r', 'r', 'd', 'L', 'r', 'u', 'r', 'u', 'l', 'l', 'd', 'l', 'l', 'd', 'R', 'd', 'd', 'r', 'd', 'l', 'l', 'd', 'l', 'l', 'u', 'R', 'u', 'R', 'l', 'd', 'l', 'd', 'r', 'r', 'U', 'r', 'U', 'r', 'd', 'l', 'l', 'l', 'u', 'R', 'l', 'd', 'r', 'r', 'r', 'u', 'L', 'U', 'd', 'r', 'd', 'l', 'l', 'l', 'u', 'R', 'l', 'd', 'r', 'r', 'r', 'u', 'L', 'u', 'U', 'l', 'u', 'R', 'R', 'R', 'l', 'l', 'l', 'd', 'r', 'd', 'd', 'r', 'd', 'l', 'l', 'l', 'u', 'R', 'l', 'd', 'r', 'r', 'r', 'u', 'L', 'u', 'u', 'l', 'u', 'r', 'r', 'r', 'd', 'L', 'r', 'u', 'l', 'l', 'D', 'D', 'D', 'r', 'd', 'L', 'L', 'd', 'l', 'l', 'u', 'R', 'u', 'R', 'l', 'd', 'l', 'd', 'r', 'r', 'U', 'r', 'U', 'r', 'd', 'l', 'l', 'l', 'u', 'R', 'l', 'd', 'r', 'r', 'r', 'u', 'L', 'U', 'd', 'r', 'd', 'l', 'l', 'l', 'u', 'R', 'l', 'd', 'r', 'r', 'r', 'u', 'L', 'u', 'U', 'l', 'u', 'R', 'l', 'd', 'r', 'd', 'd', 'r', 'd', 'l', 'l', 'l', 'u', 'R', 'l', 'd', 'r', 'r', 'U', 'U', 'U', 'r', 'r', 'u', 'u', 'l', 'D', 'r', 'd', 'L', 'r', 'u', 'l', 'L', 'D', 'D', 'D', 'r', 'd', 'L', 'L', 'r', 'r', 'u', 'l', 'l', 'l', 'D', 'r', 'd', 'L'] | 291 |
| 16 |  | Không tìm được trên máy của em | 0 |
| 17 |  | Không tìm được trên máy của em | 0 |
| 18 |  | Không tìm được trên máy của em | 0 |

## Breath first search (BFS)

|  |  |  |  |
| --- | --- | --- | --- |
| Level | Thời gian (giây) | Lời giải | Số bước đi |
| 1 | 0.12 | ['r', 'U', 'U', 'd', 'R', 'd', 'r', 'U', 'U', 'l', 'u', 'L'] | 12 |
| 2 | 0.01 | ['U', 'U', 'd', 'd', 'r', 'r', 'r', 'U', 'U'] | 9 |
| 3 | 0.24 | ['L', 'r', 'd', 'r', 'd', 'd', 'D', 'L', 'd', 'l', 'l', 'U', 'U', 'd', 'R'] | 15 |
| 4 | 0.01 | ['l', 'l', 'l', 'd', 'R', 'R', 'R'] | 7 |
| 5 |  | Không tìm được trên máy của em | 0 |
| 6 | 0.02 | ['d', 'l', 'l', 'u', 'R', 'd', 'r', 'U', 'U', 'U', 'd', 'd', 'r', 'r', 'u', 'u', 'u', 'l', 'L'] | 19 |
| 7 | 1.18 | ['L', 'U', 'U', 'U', 'l', 'u', 'R', 'd', 'd', 'd', 'd', 'L', 'd', 'l', 'U', 'U', 'U', 'U', 'l', 'u', 'R'] | 21 |
| 8 | 0.29 | ['l', 'l', 'D', 'D', 'D', 'D', 'D', 'D', 'l', 'd', 'd', 'd', 'r', 'r', 'u', 'u', 'L', 'u', 'u', 'u', 'u', 'u', 'u', 'u', 'r', 'r', 'd', 'L', 'u', 'l', 'D', 'D', 'D', 'D', 'D', 'D', 'l', 'l', 'l', 'd', 'd', 'r', 'r', 'U', 'd', 'l', 'l', 'u', 'u', 'r', 'R', 'd', 'd', 'd', 'r', 'r', 'u', 'u', 'L', 'U', 'U', 'U', 'U', 'U', 'U', 'l', 'u', 'R', 'd', 'd', 'd', 'd', 'd', 'd', 'd', 'r', 'd', 'd', 'l', 'l', 'u', 'U', 'd', 'l', 'l', 'u', 'u', 'r', 'R', 'd', 'r', 'U', 'U', 'U', 'U', 'U', 'U'] | 97 |
| 9 | 0.02 | ['d', 'l', 'U', 'd', 'l', 'l', 'U', 'R'] | 8 |
| 10 | 0.02 | ['d', 'l', 'U', 'r', 'r', 'r', 'd', 'L', 'u', 'l', 'l', 'd', 'd', 'r', 'U', 'l', 'u', 'R', 'u', 'u', 'l', 'D', 'r', 'd', 'd', 'r', 'r', 'u', 'L', 'd', 'l', 'U', 'U'] | 33 |
| 11 | 0.02 | ['r', 'D', 'l', 'd', 'd', 'r', 'r', 'r', 'u', 'u', 'L', 'L', 'r', 'r', 'd', 'd', 'l', 'l', 'U', 'd', 'r', 'r', 'u', 'u', 'l', 'u', 'l', 'l', 'D', 'R', 'd', 'd', 'l', 'U'] | 34 |
| 12 | 0.13 | ['r', 'r', 'D', 'u', 'l', 'l', 'd', 'R', 'R', 'u', 'r', 'D', 'r', 'd', 'd', 'l', 'l', 'U', 'R', 'u', 'u', 'l', 'D'] | 23 |
| 13 | 0.2 | ['d', 'r', 'r', 'd', 'r', 'r', 'u', 'L', 'L', 'u', 'u', 'r', 'D', 'D', 'u', 'l', 'l', 'l', 'd', 'R', 'l', 'd', 'd', 'r', 'r', 'u', 'U', 'r', 'r', 'd', 'L'] | 31 |
| 14 | 3.81 | ['u', 'l', 'l', 'D', 'L', 'd', 'R', 'u', 'u', 'r', 'r', 'd', 'L', 'L', 'r', 'r', 'd', 'd', 'l', 'U', 'r', 'u', 'L'] | 23 |
| 15 | 0.37 | ['r', 'd', 'd', 'L', 'r', 'u', 'u', 'l', 'D', 'l', 'D', 'D', 'D', 'r', 'd', 'L', 'L', 'd', 'l', 'l', 'u', 'R', 'u', 'R', 'l', 'd', 'd', 'r', 'U', 'r', 'U', 'U', 'U', 'l', 'u', 'R', 'R', 'R', 'd', 'L', 'u', 'l', 'D', 'D', 'D', 'r', 'd', 'L', 'L', 'd', 'l', 'l', 'u', 'R', 'u', 'R', 'l', 'd', 'd', 'r', 'U', 'r', 'U', 'U', 'U', 'l', 'u', 'R', 'd', 'd', 'd', 'd', 'l', 'l', 'u', 'R', 'd', 'r', 'U', 'U', 'U', 'r', 'r', 'u', 'u', 'l', 'D', 'r', 'd', 'L', 'u', 'L', 'D', 'D', 'D', 'r', 'd', 'L', 'L', 'u', 'l', 'D', 'r', 'd', 'L'] | 105 |
| 16 | 28.73 | ['R', 'u', 'r', 'r', 'd', 'd', 'd', 'd', 'l', 'D', 'R', 'u', 'u', 'u', 'u', 'L', 'L', 'L', 'r', 'd', 'R', 'D', 'r', 'd', 'd', 'l', 'L', 'd', 'l', 'l', 'U', 'U', 'd', 'R'] | 34 |
| 17 |  | Không tìm được trên máy của em | 0 |
| 18 |  | Không tìm được trên máy của em | 0 |

## Uniform cost search (UCS)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Level | Thời gian (giây) | Lời giải | Số bước đi | Chi phí |
| 1 | 0.08 | ['r', 'U', 'R', 'd', 'r', 'U', 'U', 'l', 'L', 'd', 'l', 'U'] | 12 | 6 |
| 2 | 0.01 | ['U', 'U', 'd', 'd', 'r', 'r', 'r', 'U', 'U'] | 9 | 5 |
| 3 | 0.12 | ['L', 'r', 'd', 'r', 'd', 'd', 'D', 'L', 'd', 'l', 'l', 'U', 'U', 'd', 'R'] | 15 | 9 |
| 4 | 0 | ['l', 'l', 'l', 'd', 'R', 'R', 'R'] | 7 | 4 |
| 5 | 91.14 | ['u', 'l', 'u', 'R', 'R', 'd', 'r', 'U', 'U', 'U', 'd', 'l', 'L', 'U', 'd', 'L', 'd', 'l', 'U', 'U'] | 20 | 10 |
| 6 | 0.02 | ['d', 'l', 'l', 'u', 'R', 'd', 'r', 'U', 'U', 'U', 'd', 'd', 'r', 'r', 'u', 'u', 'u', 'l', 'L'] | 19 | 14 |
| 7 | 0.75 | ['L', 'U', 'U', 'U', 'l', 'u', 'R', 'd', 'd', 'd', 'd', 'L', 'd', 'l', 'U', 'U', 'U', 'U', 'l', 'u', 'R'] | 21 | 10 |
| 8 | 0.29 | ['l', 'l', 'D', 'D', 'D', 'D', 'D', 'D', 'l', 'd', 'd', 'd', 'r', 'r', 'u', 'u', 'L', 'u', 'u', 'u', 'u', 'u', 'u', 'u', 'r', 'r', 'd', 'L', 'u', 'l', 'D', 'D', 'D', 'D', 'D', 'D', 'l', 'l', 'l', 'd', 'd', 'r', 'r', 'U', 'd', 'l', 'l', 'u', 'u', 'r', 'R', 'd', 'd', 'd', 'r', 'r', 'u', 'u', 'L', 'U', 'U', 'U', 'U', 'U', 'U', 'l', 'u', 'R', 'd', 'd', 'd', 'd', 'd', 'd', 'd', 'r', 'd', 'd', 'l', 'l', 'u', 'U', 'd', 'l', 'l', 'u', 'u', 'r', 'R', 'd', 'r', 'U', 'U', 'U', 'U', 'U', 'U'] | 97 | 65 |
| 9 | 0.01 | ['d', 'l', 'U', 'd', 'l', 'l', 'U', 'R'] | 8 | 5 |
| 10 | 0.02 | ['d', 'l', 'U', 'r', 'r', 'r', 'd', 'L', 'u', 'l', 'l', 'd', 'd', 'r', 'U', 'l', 'u', 'R', 'u', 'u', 'l', 'D', 'r', 'd', 'd', 'r', 'r', 'u', 'L', 'd', 'l', 'U', 'U'] | 33 | 25 |
| 11 | 0.02 | ['r', 'D', 'l', 'd', 'd', 'r', 'r', 'r', 'u', 'u', 'L', 'L', 'r', 'r', 'd', 'd', 'l', 'l', 'U', 'd', 'r', 'r', 'u', 'u', 'l', 'u', 'l', 'l', 'D', 'R', 'd', 'd', 'l', 'U'] | 34 | 27 |
| 12 | 0.15 | ['r', 'r', 'D', 'u', 'l', 'l', 'd', 'R', 'R', 'u', 'r', 'D', 'r', 'd', 'd', 'l', 'l', 'U', 'R', 'u', 'u', 'l', 'D'] | 23 | 16 |
| 13 | 0.26 | ['d', 'r', 'r', 'd', 'r', 'r', 'u', 'L', 'L', 'u', 'u', 'r', 'D', 'D', 'u', 'l', 'l', 'l', 'd', 'R', 'l', 'd', 'd', 'r', 'r', 'u', 'U', 'r', 'r', 'd', 'L'] | 31 | 24 |
| 14 | 4.23 | ['u', 'l', 'l', 'D', 'L', 'd', 'R', 'u', 'u', 'r', 'r', 'd', 'L', 'L', 'r', 'r', 'd', 'd', 'l', 'U', 'r', 'u', 'L'] | 23 | 16 |
| 15 | 0.4 | ['r', 'd', 'd', 'L', 'r', 'u', 'u', 'l', 'D', 'l', 'D', 'D', 'D', 'r', 'd', 'L', 'L', 'd', 'l', 'l', 'u', 'R', 'u', 'R', 'l', 'd', 'd', 'r', 'U', 'r', 'U', 'U', 'U', 'l', 'u', 'R', 'R', 'R', 'd', 'L', 'u', 'l', 'D', 'D', 'D', 'r', 'd', 'L', 'L', 'd', 'l', 'l', 'u', 'R', 'u', 'R', 'l', 'd', 'd', 'r', 'U', 'r', 'U', 'U', 'U', 'l', 'u', 'R', 'd', 'd', 'd', 'd', 'l', 'l', 'u', 'R', 'd', 'r', 'U', 'U', 'U', 'r', 'r', 'u', 'u', 'l', 'D', 'r', 'd', 'L', 'u', 'L', 'D', 'D', 'D', 'r', 'd', 'L', 'L', 'u', 'l', 'D', 'r', 'd', 'L'] | 105 | 62 |
| 16 | 21.77 | ['R', 'u', 'r', 'r', 'd', 'd', 'd', 'd', 'l', 'D', 'R', 'u', 'u', 'u', 'u', 'L', 'L', 'L', 'r', 'd', 'R', 'D', 'r', 'd', 'd', 'l', 'L', 'd', 'l', 'l', 'U', 'U', 'd', 'R'] | 34 | 22 |
| 17 |  | Không tìm được trên máy của em | 0 | 0 |
| 18 |  | Không tìm được trên máy của em | 0 | 0 |

## Độ dài đường đi của 3 thuật toán

|  |  |  |  |
| --- | --- | --- | --- |
| Level | DFS | BFS | UCS |
| 1 | 79 | 12 | 12 |
| 2 | 24 | 9 | 9 |
| 3 | 403 | 15 | 15 |
| 4 | 27 | 7 | 7 |
| 5 | 0 | 0 | 20 |
| 6 | 55 | 19 | 19 |
| 7 | 707 | 21 | 21 |
| 8 | 323 | 97 | 97 |
| 9 | 74 | 8 | 8 |
| 10 | 37 | 33 | 33 |
| 11 | 36 | 34 | 34 |
| 12 | 109 | 23 | 23 |
| 13 | 185 | 31 | 31 |
| 14 | 865 | 23 | 23 |
| 15 | 291 | 105 | 105 |
| 16 | 0 | 34 | 34 |
| 17 | 0 | 0 | 0 |
| 18 | 0 | 0 | 0 |

## Nhận xét

* Lời giải 2 thuật toán BFS và UCS tìm được có độ dài như nhau và ngắn hơn đáng kể so với lời giải mà thuật toán DFS tìm được.
* Thuật toán UCS là tốt nhất trong 3 thuật toán vì có thể tìm ra nhiều lời giải nhất với cùng lượng tài nguyên.
* Các màn khó:
  + Màn 5 có không gian tìm kiếm rất lớn.
  + Màn 17, 18 có thiết kế phức tạp.