Sliding Puzzle

From : Leetcode
Difficulty: Hard
By : Aung Khant Oo

Problem Statement

773. Sliding Puzzle

Hard 907 30 Add to List Share

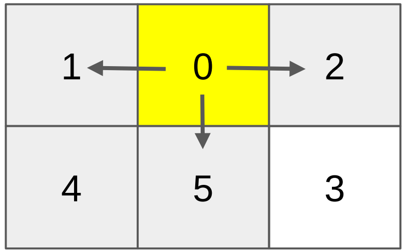
On a 2x3 `board`, there are 5 tiles represented by the integers 1 through 5, and an empty square represented by 0.

A move consists of choosing `0` and a 4-directionally adjacent number and swapping it.

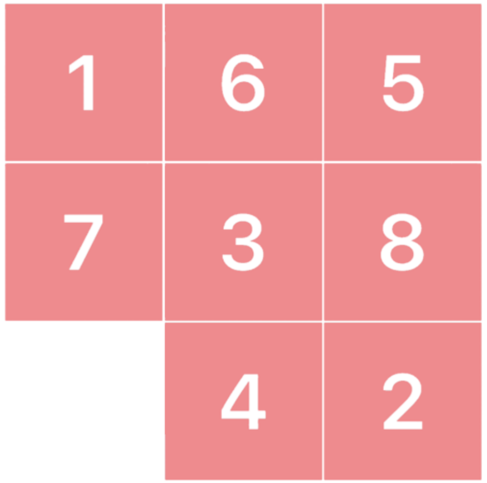
The state of the board is *solved* if and only if the `board` is `[[1,2,3],[4,5,0]].`

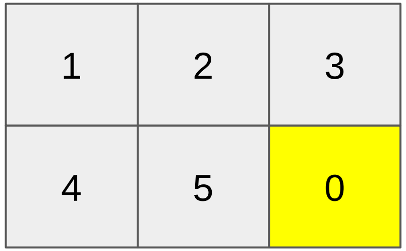
Given a puzzle board, return the least number of moves required so that the state of the board is solved. If it is impossible for the state of the board to be solved, return -1.

Examples:



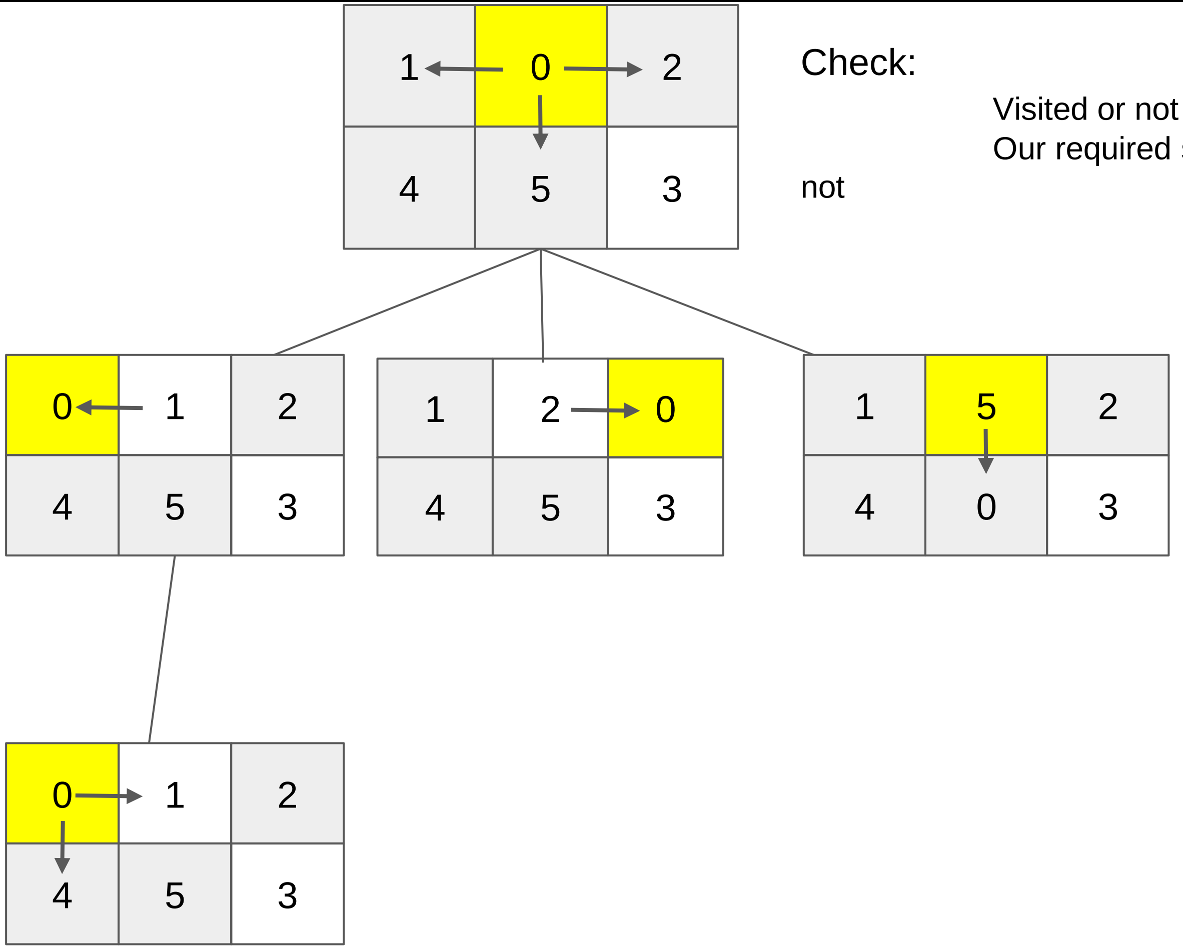
【
【1, 0, 2】
【4, 5, 3】
】





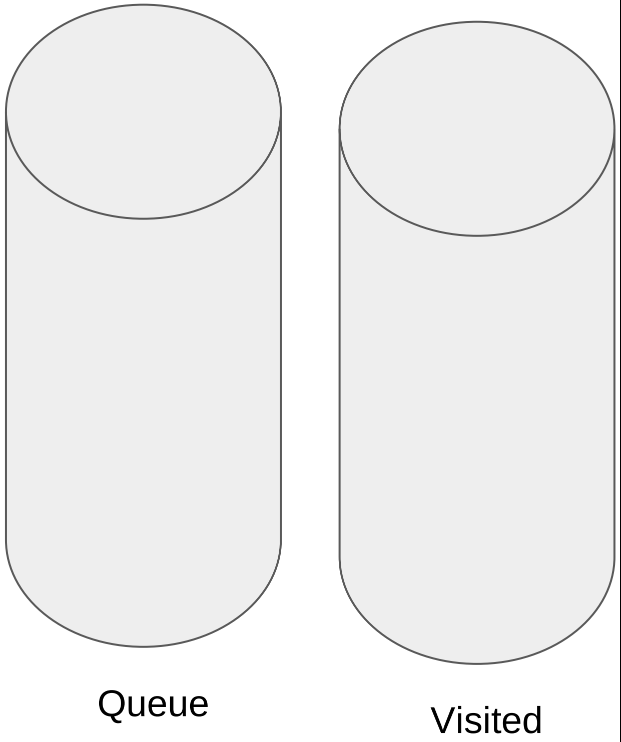
【
【1, 2, 3】
【4, 5, 0】
】

BFS(Breadth first search)

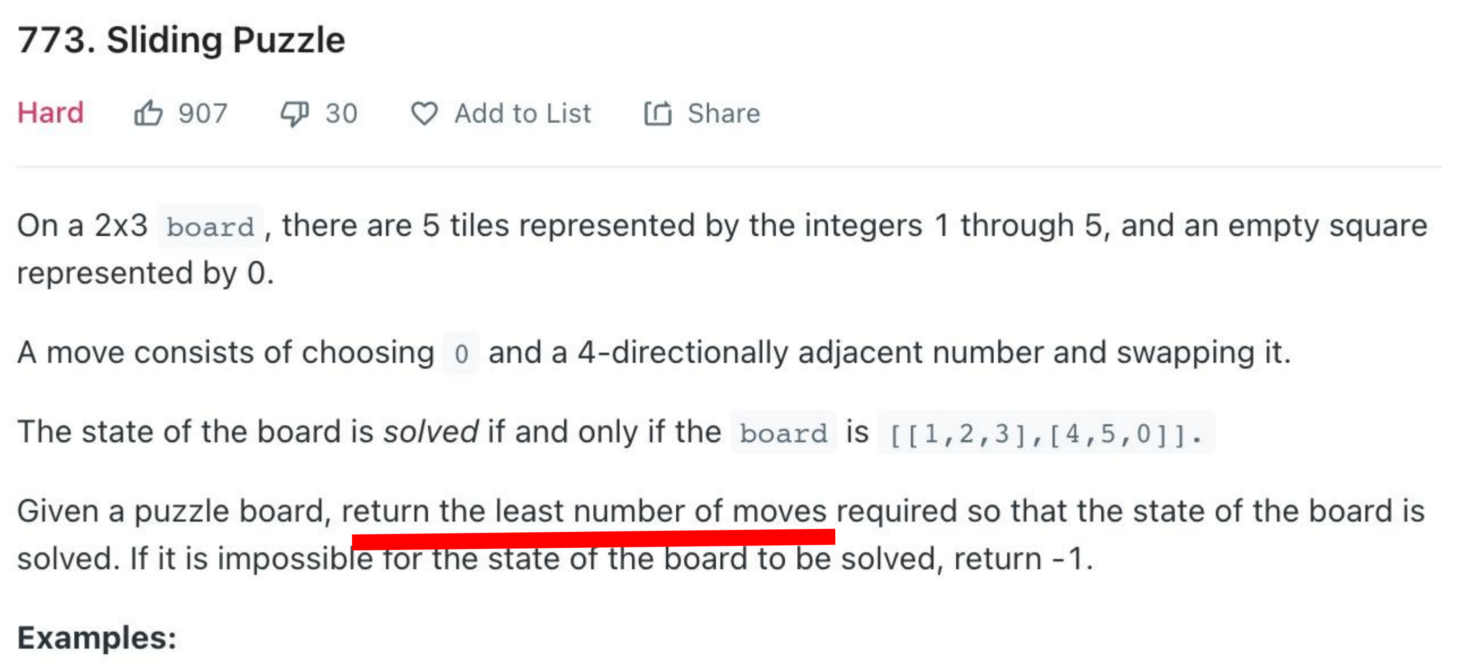


Check:

Visited or not
Our required state or
not



Why BFS???



773. Sliding Puzzle

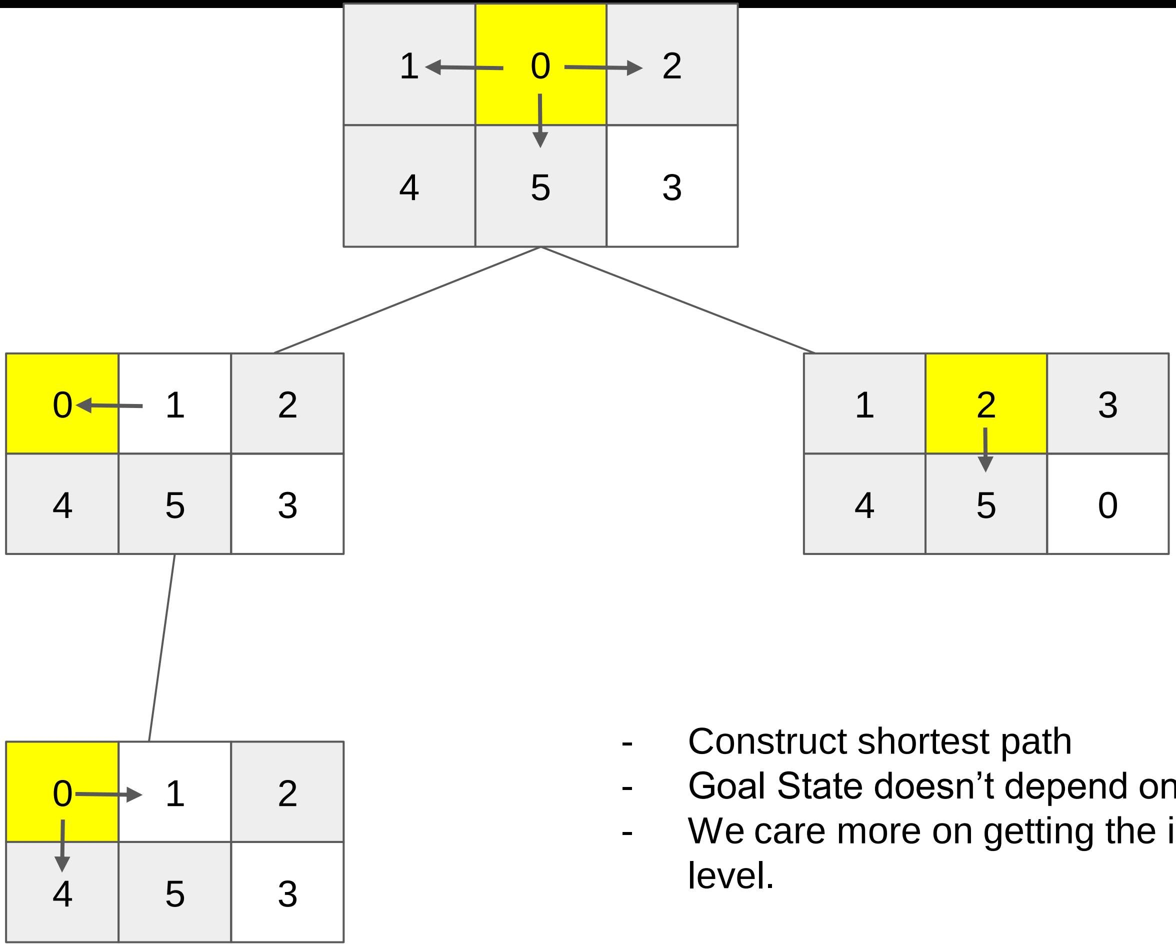
On a 2x3 `board`, there are 5 tiles represented by the integers 1 through 5, and an empty square represented by 0.

A move consists of choosing `0` and a 4-directionally adjacent number and swapping it.

The state of the board is *solved* if and only if the `board` is `[[1,2,3],[4,5,0]].`

Given a puzzle board, return the least number of moves required so that the state of the board is solved. If it is impossible for the state of the board to be solved, return -1.

Examples:



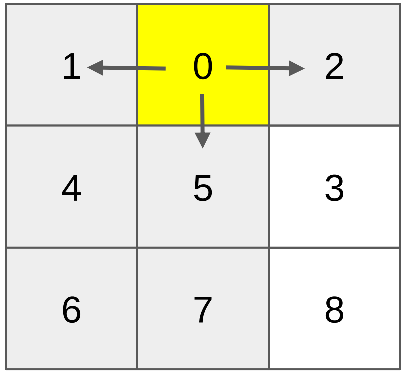
- Construct shortest path
- Goal State doesn’t depend on how deep it go.
- We care more on getting the ideal state in this level.

Optimization(A*)

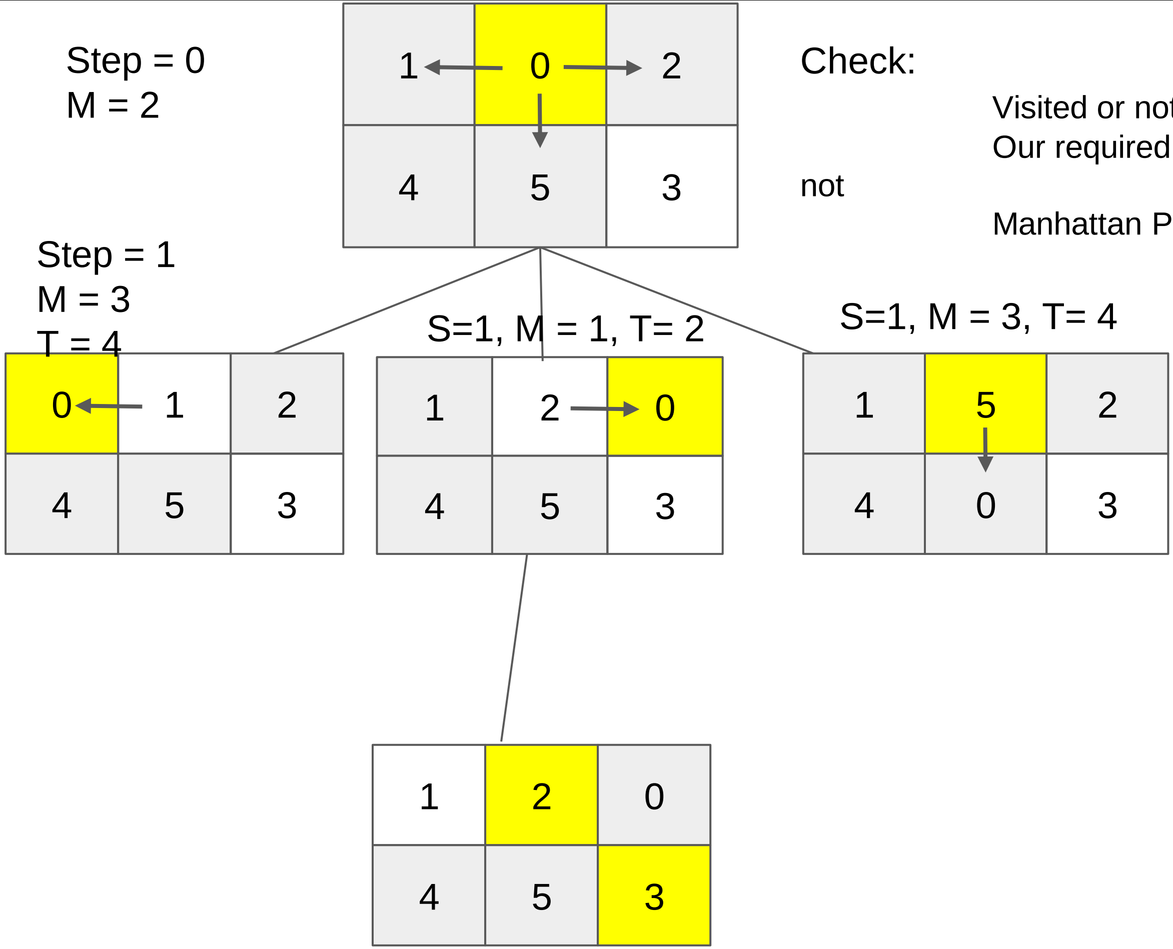
8-Puzzle

9! = 362,880 States

O(2^n)



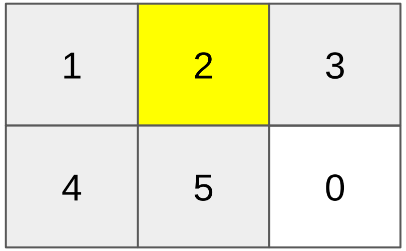
Step = 0
M = 2



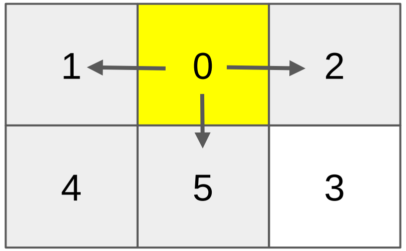
Check:

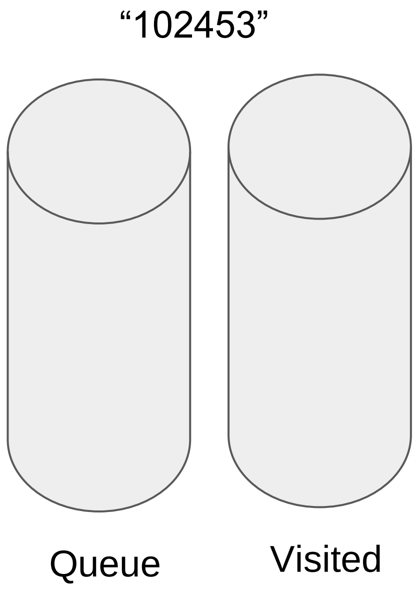
Visited or not
Our required state or not

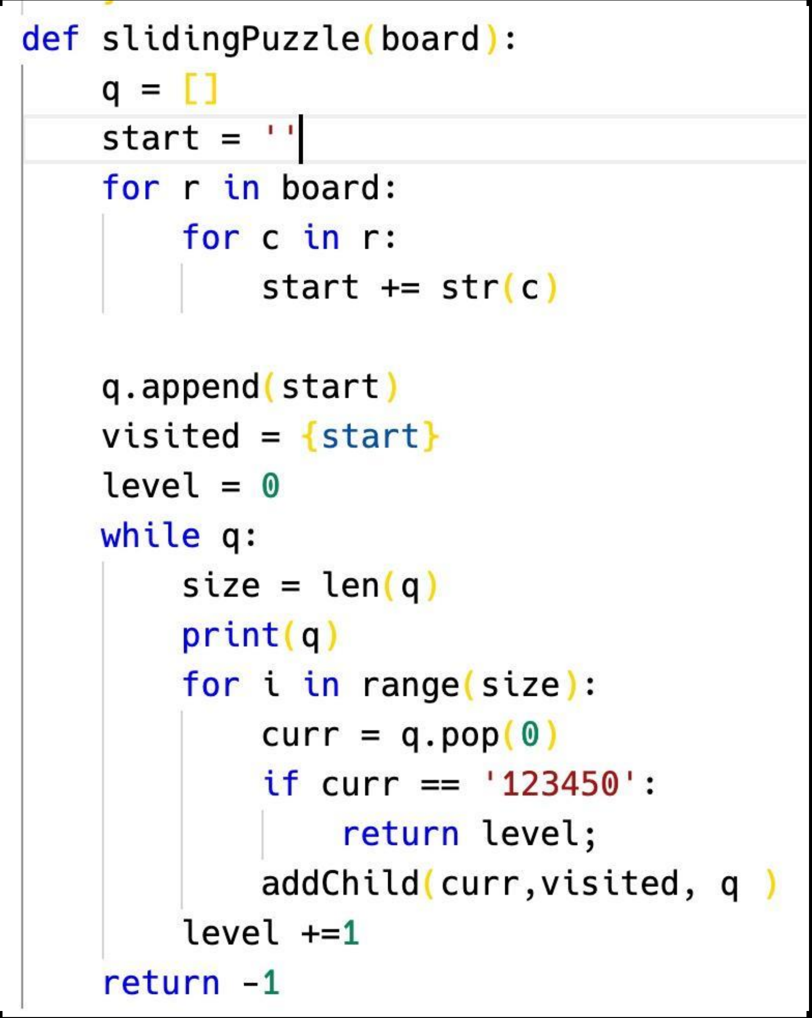
Manhattan Priority



Code







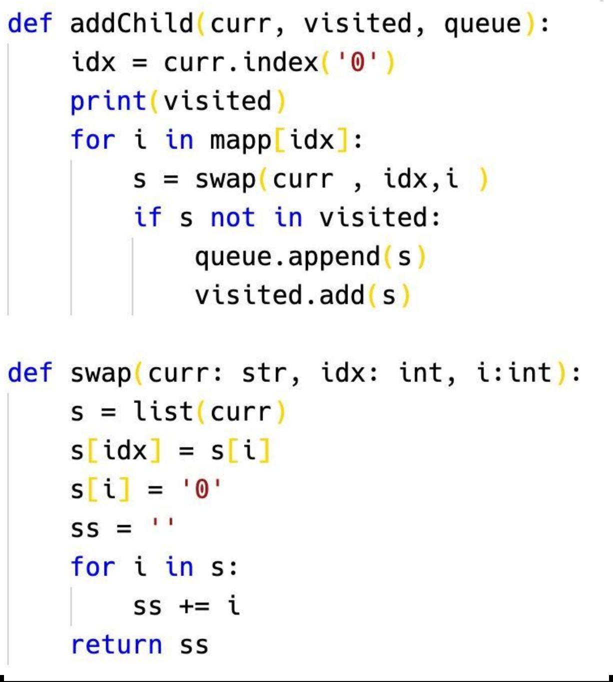
```python
def slidingPuzzle(board):
    q = []
    start = ''
    for r in board:
        for c in r:
            start += str(c)

    q.append(start)
    visited = {start}
    level = 0
    while q:
        size = len(q)
        print(q)
        for i in range(size):
            curr = q.pop(0)
            if curr == '123450':
                return level;
            addChild(curr,visited, q )
        level +=1
    return -1
```

Code

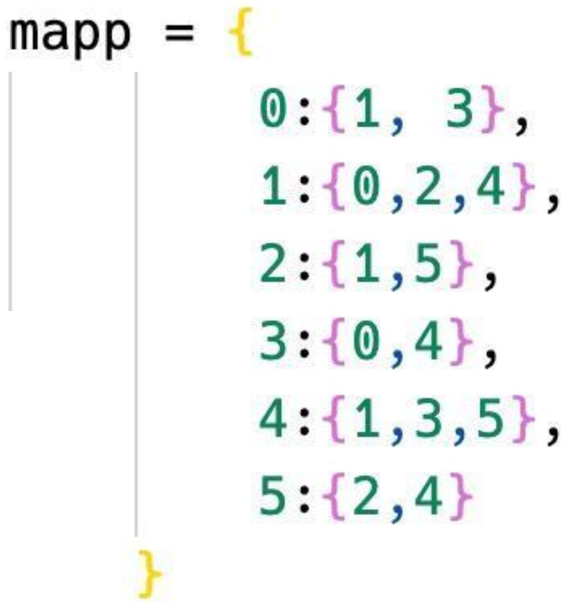
1	2	3
4	5	0

“102
453”

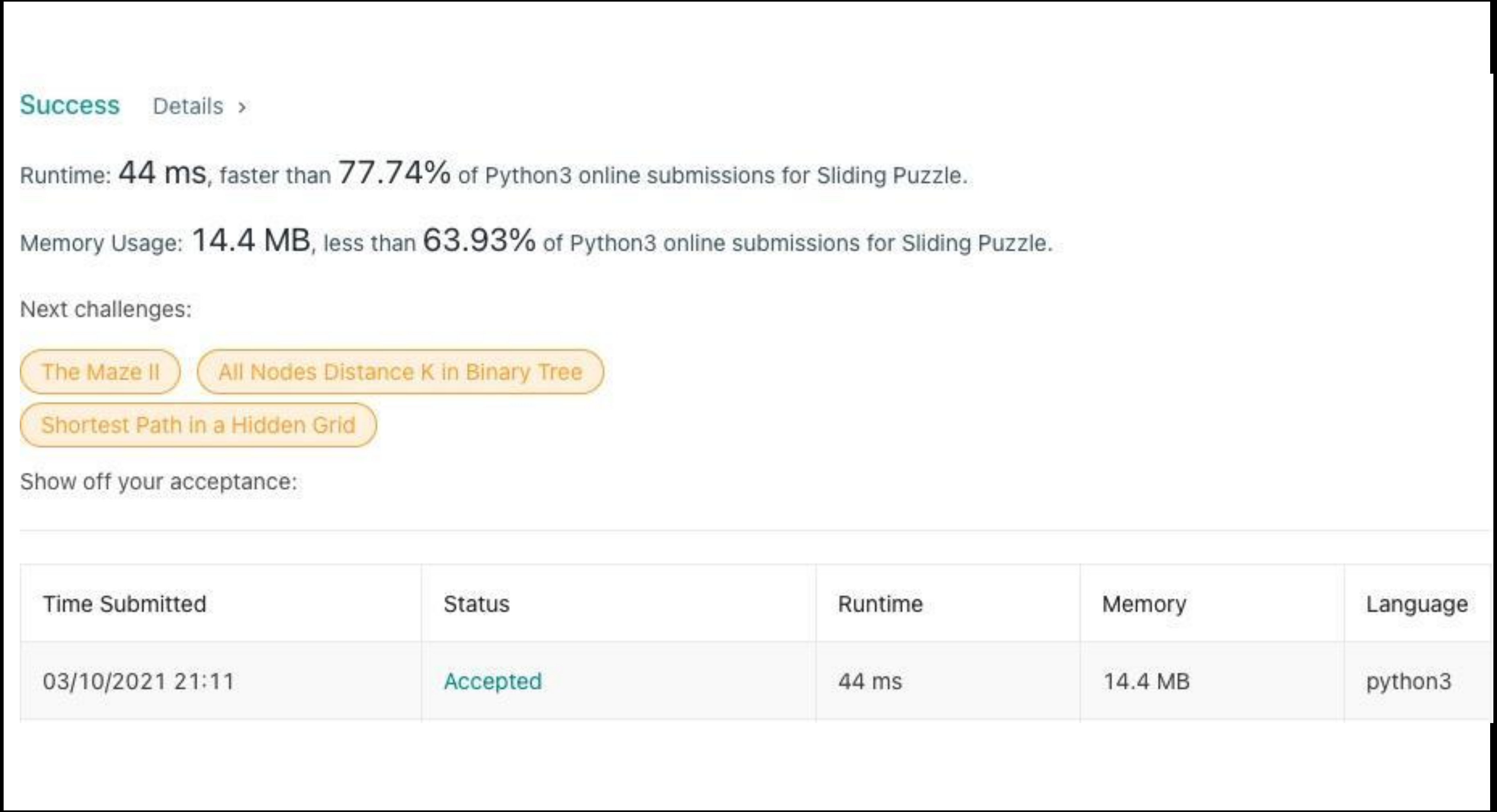


```python
def addChild(curr, visited, queue):
    idx = curr.index('0')
    print(visited)
    for i in mapp[idx]:
        s = swap(curr , idx,i )
        if s not in visited:
            queue.append(s)
            visited.add(s)

def swap(curr: str, idx: int, i:int):
    s = list(curr)
    s[idx] = s[i]
    s[i] = '0'
    ss = ''
    for i in s:
        ss += i
    return ss
```



```python
mapp = {
            0:{1, 3},
            1:{0,2,4},
            2:{1,5},
            3:{0,4},
            4:{1,3,5},
            5:{2,4}
      }
```



Success Details ›

Runtime: 44 ms, faster than 77.74% of Python3 online submissions for Sliding Puzzle.

Memory Usage: 14.4 MB, less than 63.93% of Python3 online submissions for Sliding Puzzle.

Next challenges:

The Maze II

All Nodes Distance K in Binary Tree

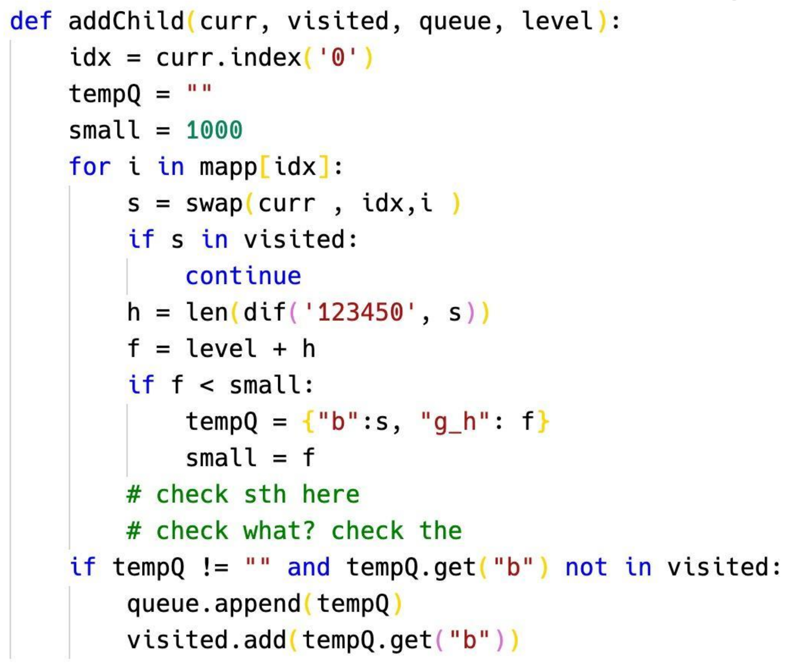
Shortest Path in a Hidden Grid

Show off your acceptance:

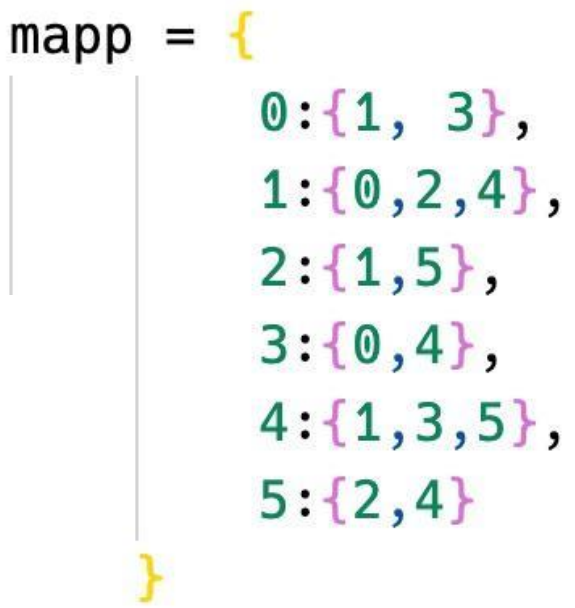
Time Submitted	Status	Runtime	Memory	Language
03/10/2021 21:11	Accepted	44 ms	14.4 MB	python3

A* Code

1	2	3
4	5	0



```python
def addChild(curr, visited, queue, level):
    idx = curr.index('0')
    tempQ = ""
    small = 1000
    for i in mapp[idx]:
        s = swap(curr , idx,i )
        if s in visited:
            continue
        h = len(dif('123450', s))
        f = level + h
        if f < small:
            tempQ = {"b":s, "g_h": f}
            small = f
        # check sth here
        # check what? check the
    if tempQ != "" and tempQ.get("b") not in visited:
        queue.append(tempQ)
        visited.add(tempQ.get("b"))
```



```python
mapp = {
        0:{1, 3},
        1:{0,2,4},
        2:{1,5},
        3:{0,4},
        4:{1,3,5},
        5:{2,4}
}
```

Thank you

References

- https://www.youtube.com/watch?v=YP9ElstYN-k&t=1s&ab_channel=GregoryLi
- https://www.youtube.com/watch?v=GuCzYxHa7iA&ab_channel=JinyueHan