

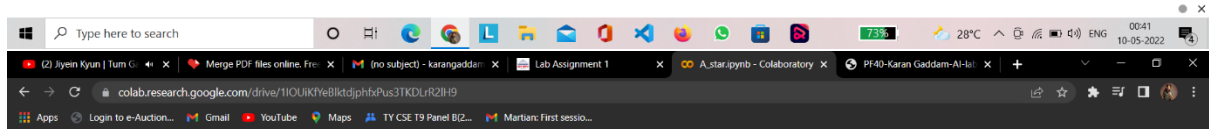
```
+ Code + Text
import numpy as np

[ ] class Puzzle:
    def __init__(self):
        self.matrix = []
        f_of_n = 999
        g_of_n = 999
        h_of_n = 999
        parent_index = 999
        self_index = 999

[ ] def HurasticCost(A,B):
    count = 0
    for i in range(len(A.matrix)):
        for j in range(len(A.matrix[i])):
            if A.matrix[i][j] != B.matrix[i][j]:
                count += 1
    return count

[ ] def CopyMatrix_3d(A,B):
    for i in range(3):
        temp = []
        for j in range(3):
            temp.append(B[i][j])
        A.append(temp)

[ ] def index_2d(myList, v):
    for i, x in enumerate(myList):
        if v in x:
            return i, x.index(v)
```

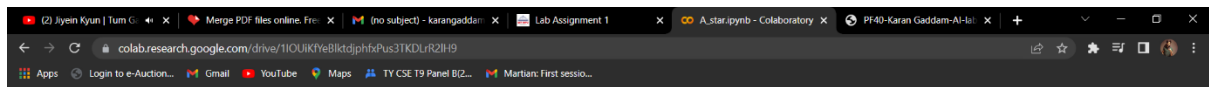


```
[ ] def index_2d(myList, v):
    for i, x in enumerate(myList):
        if v in x:
            return i, x.index(v)

[ ] def ExchangePlaces(A,B):
    variable_index = index_2d(A.matrix,0)
    variable = A.matrix[B[0]][B[1]]
    A.matrix[B[0]][B[1]] = 0
    A.matrix[variable_index[0]][variable_index[1]] = variable

[ ] def PossibleWays(A):
    choices = []
    index = index_2d(A.matrix,0)
    if index[0] + 1 < 3:
        choices.append([index[0] + 1, index[1]])
    if index[0] - 1 >= 0:
        choices.append([index[0] - 1, index[1]])
    if index[1] + 1 < 3:
        choices.append([index[0], index[1] + 1])
    if index[1] - 1 >= 0:
        choices.append([index[0], index[1] - 1])
    return choices

[ ] def ReturnMin(A):
    min_value = 999
    temp = Puzzle()
    for i in A:
```

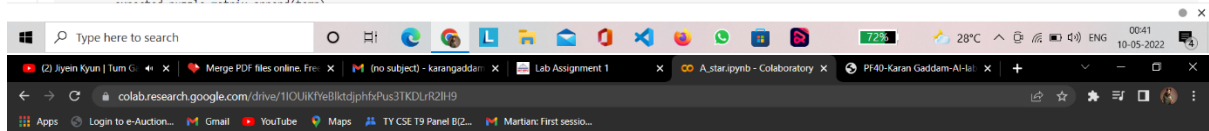


```
A_star.ipynb
File Edit View Insert Runtime Tools Help Last edited on 4 May
+ Code + Text
[ ] def ReturnMin(A):
    min_value = 999
    temp = Puzzle()
    for i in A:
        if i.f_of_n <= min_value:
            min_value = i.f_of_n
            temp = i
    return temp

[ ] puzzle = Puzzle()
print("Enter the puzzle:")
for i in range(3):
    temp = input()
    temp = list(temp.split(" "))
    temp = [int(x) for x in temp]
    puzzle.matrix.append(temp)
print(puzzle.matrix)

Enter the puzzle:
1 2 3
0 4 6
7 5 8
[[1, 2, 3], [0, 4, 6], [7, 5, 8]]

[ ] expected_puzzle = Puzzle()
print("Enter the puzzle:")
for i in range(3):
    temp = input()
    temp = list(temp.split(" "))
    temp = [int(x) for x in temp]
```



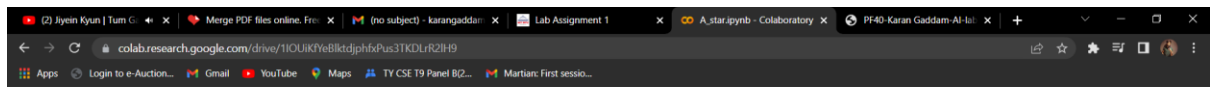
```
A_star.ipynb
File Edit View Insert Runtime Tools Help Last edited on 4 May
+ Code + Text
[ ] expected_puzzle = Puzzle()
print("Enter the puzzle:")
for i in range(3):
    temp = input()
    temp = list(temp.split(" "))
    temp = [int(x) for x in temp]
    expected_puzzle.matrix.append(temp)
print(expected_puzzle.matrix)

Enter the puzzle:
1 2 3
4 5 6
7 8 0
[[1, 2, 3], [4, 5, 6], [7, 8, 0]]

[ ] expected_puzzle.matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 0]]
puzzle.matrix = [[1, 2, 3], [0, 4, 6], [7, 5, 8]]

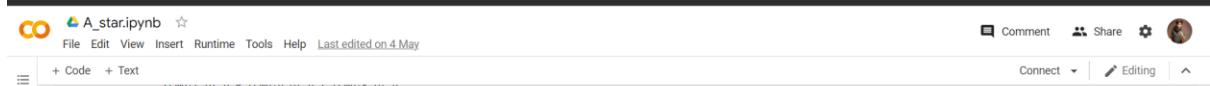
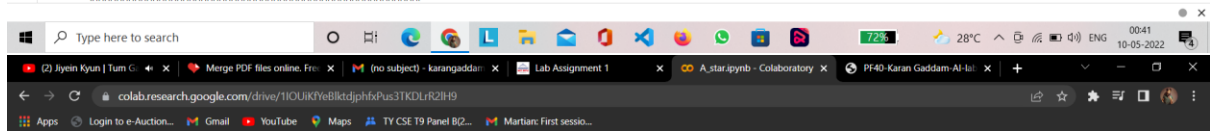
[ ] puzzle.g_of_n = 0
puzzle.h_of_n = HurasticCost(puzzle, expected_puzzle)
puzzle.f_of_n = puzzle.h_of_n + puzzle.g_of_n

[ ] all_puzzles = []
all_puzzles.append(puzzle)
level = 0
while puzzle.matrix != expected_puzzle.matrix:
    choices = Possibleways(puzzle)
    previous_index = index_2d(puzzle.matrix, 0)
    level = level + 1
```



```
A_star.ipynb
File Edit View Insert Runtime Tools Help Last edited on 4 May
+ Code + Text
[ ] all_puzzles = []
all_puzzles.append(puzzle)
level = 0
while puzzle.matrix != expected_puzzle.matrix:
    choices = Possibleways(puzzle)
    previous_index = index_2d(puzzle.matrix,0)
    level = level + 1
    for i in choices:
        if i != previous_index:
            temp = Puzzle()
            CopyMatrix_3d(temp.matrix,puzzle.matrix)
            ExchangePlaces(temp,i)
            temp.h_of_n = HurasticCost(temp,expected_puzzle)
            temp.g_of_n = level
            temp.f_of_n = temp.h_of_n + temp.g_of_n
            all_puzzles.append(temp)
    puzzle = ReturnMin(all_puzzles)
    print("#####")
    print(puzzle.matrix)
    print(f"F(n)= {puzzle.f_of_n}")
    print(f"G(n)= {puzzle.g_of_n}")
    print(f"H(n)= {puzzle.h_of_n}")

#####
[[1, 2, 3], [4, 0, 6], [7, 5, 8]]
F(n)= 4
G(n)= 1
H(n)= 3
#####
```



```
A_star.ipynb
File Edit View Insert Runtime Tools Help Last edited on 4 May
+ Code + Text
[ ] all_puzzles.append(temp)
puzzle = ReturnMin(all_puzzles)
print("#####")
print(puzzle.matrix)
print(f"F(n)= {puzzle.f_of_n}")
print(f"G(n)= {puzzle.g_of_n}")
print(f"H(n)= {puzzle.h_of_n}")

#####
[[1, 2, 3], [4, 0, 6], [7, 5, 8]]
F(n)= 4
G(n)= 1
H(n)= 3
#####
[[1, 2, 3], [4, 5, 6], [7, 0, 8]]
F(n)= 4
G(n)= 2
H(n)= 2
#####
[[1, 2, 3], [4, 5, 6], [7, 8, 0]]
F(n)= 3
G(n)= 3
H(n)= 0
#####

[ ]

[ ]
```

