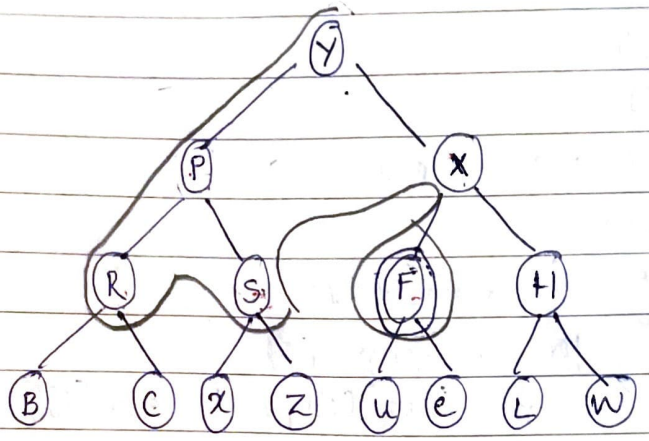
LAB - 04 15/10/24

1. Implementing iterative deepening search



S → Y G → F

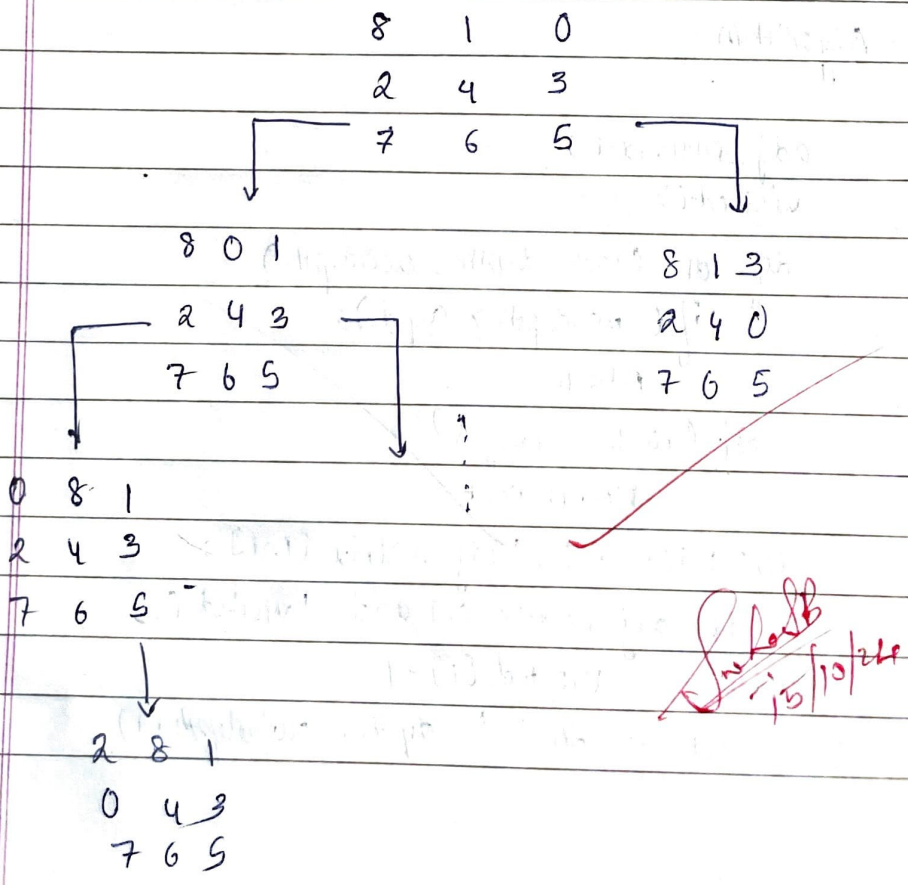
Depth or level	Traversing
0	Y
1	Y P X
2	Y P R S X F

Algorithm:

```
adj_matrix=[]
visited=[]
def dfs (root, depth, currdepth)
    if ( currdepth > depth):
        return
    def (root == target)
        return root
else: for i in adj matrix [int]:
    if adj_matrix [i] and !visited (i)
        visited [i] = 1
    return dfs ( i, depth, currdepth+1)
```

Q. A* 8 Puzzle Algorithm

```
function AStar (start_state, goal_state):
start = []
end = []
def calc_h(curr, end)
    mid = 0
    for i in range [3]:
        for j in range [3]:
            if (curr[i][j] != end[i][j])
            mid += 1
return mid
def (state):
ind_state = state.ind(3)
    min (calc_h [state [ind[0]-1] ind [1]],
        [calc_h [state [ind [0]] [ind [1]+1])
        [calc_h [state [ind [0]] [ind [1]-1]
    )
```



```
Enter the start state (9 numbers, use 0 for the blank space):
Start state: 1 2 3 8 0 4 7 6 5
Enter the goal state (9 numbers, use 0 for the blank space):
Goal state: 2 8 1 0 4 3 7 6 5
Solution found in 9 moves.
[1, 2, 3]
[8, 0, 4]
[7, 6, 5]

[1, 0, 3]
[8, 2, 4]
[7, 6, 5]

[0, 1, 3]
[8, 2, 4]
[7, 6, 5]

[8, 1, 3]
[0, 2, 4]
[7, 6, 5]

[8, 1, 3]
[2, 0, 4]
[7, 6, 5]

[8, 1, 3]
[2, 4, 0]
[7, 6, 5]

[8, 1, 0]
[2, 4, 3]
[7, 6, 5]

[8, 0, 1]
[2, 4, 3]
[7, 6, 5]

[0, 8, 1]
[2, 4, 3]
[7, 6, 5]
```