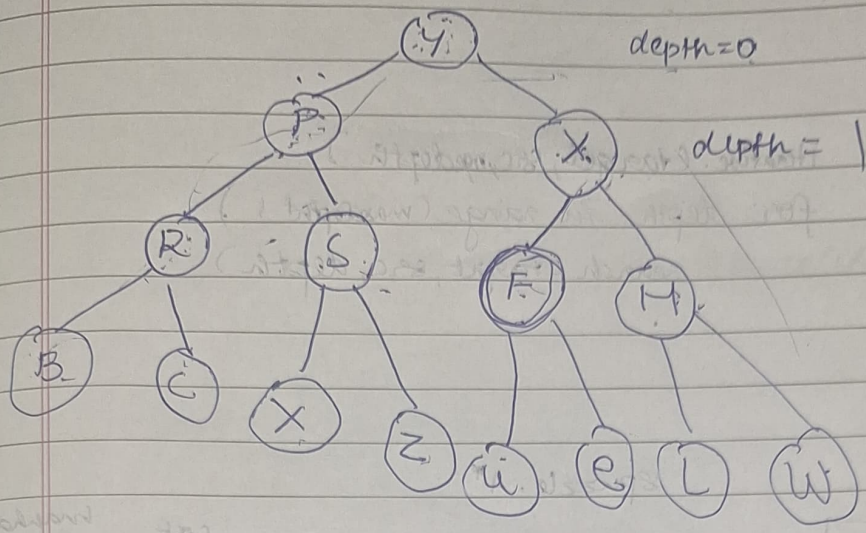
15/10/24

Solving 8puzzle ~~problem~~ using A* algorithm & Iterative deepening.



The depth is 0 so it will start at Y
Y is not goal state.

The depth is incremented by 1. It will the path will be
Y → P → X
It didn't reach goal state.

The depth is incremented by 1. It will traverse uptill the end of left node and visit the child node of parent node.
It's a combination DFS and BFS.
The path will be
Y → P → R → S → X → F
It reached the goal state.

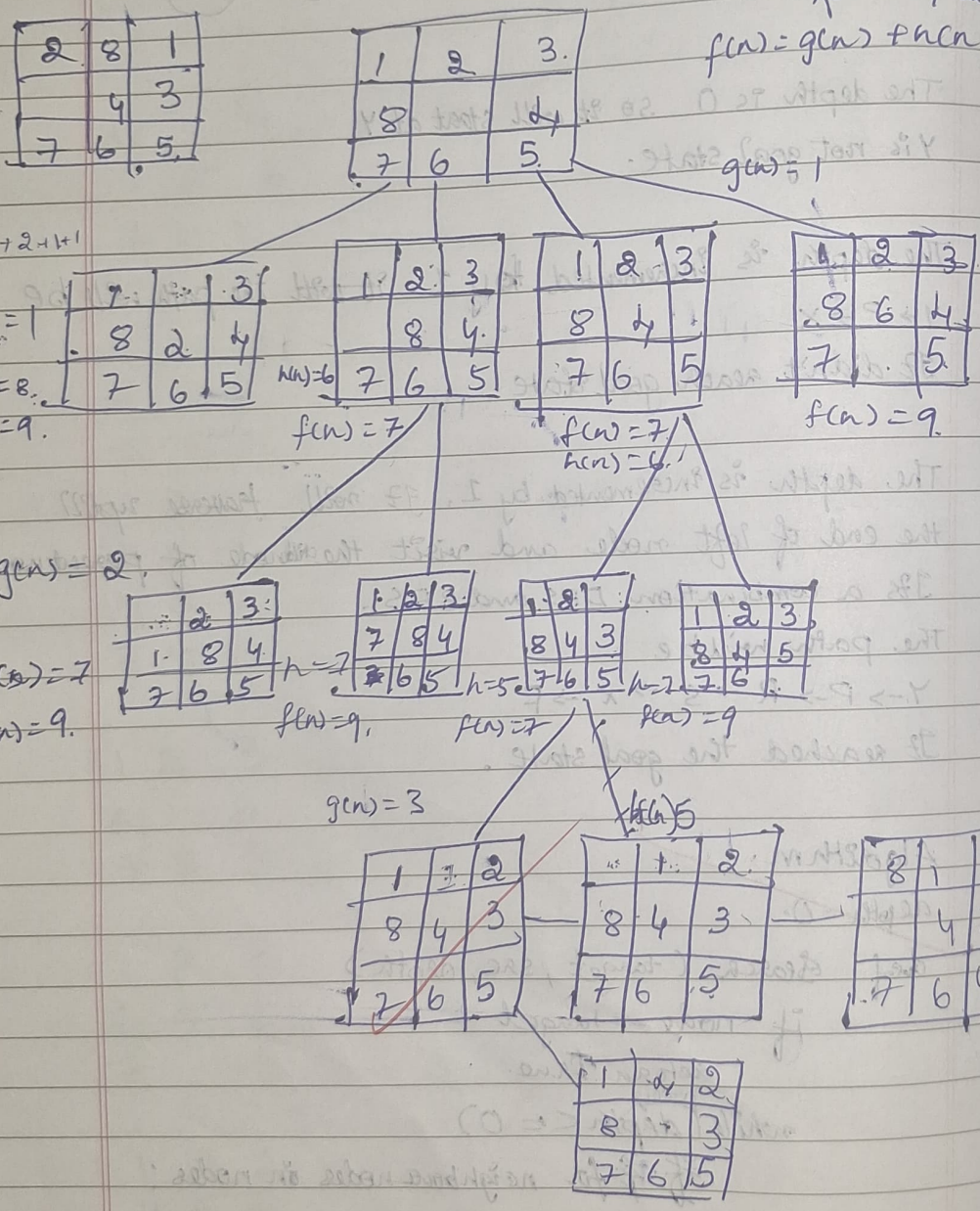
Algorithm:
depth = 0
def Search (target, src, depth)
    if node == target
        return True.
    while [depth <= 0]
        if for neighbour nodes in nodes:

search (target, neighbours nodes, depth-1)

def iterative (target, src, maxdepth)
    for depth in range (maxdepth+1)
        search (target, src, depth)

8puzzle A*

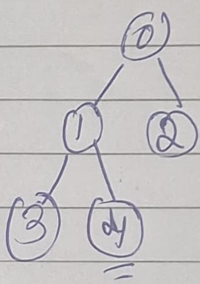
cost ↑ g(n) = g(n) + h(n) ↑ manhattan distance



A* algorithm.

```
def Astart (startstate, goalstate):
    cost = 0
if (startstate != visited[]).
states[]=generationmoves(startstate)
    for states in generation states:
        f(n) = cost + manhattendistance(state)

    sort(f(n)).
    visited.append(state).
    A_start(state, goalstate).
```



Output for IDS,

Iteration 1

0 -> 1 -> 2.

Iteration 2.

0 -> 1 -> 3 -> 4