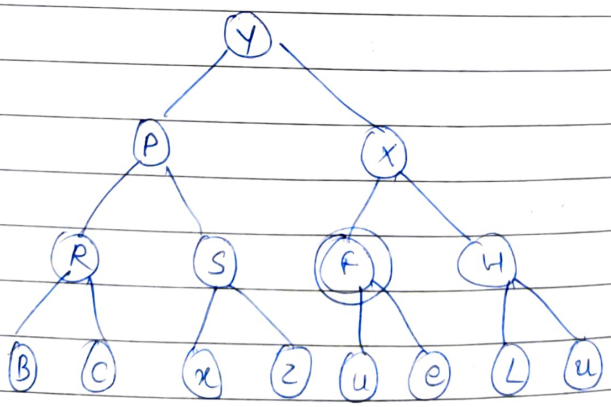
15/10/24

Recursive

→ Iterative Deepening Depth First Search

pseudo code:

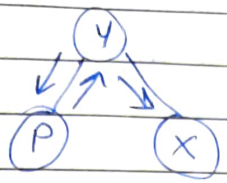
```
function IDDFS (root, goal):
    for d = 0 to INT_MAX
        result = DLS (root, goal, depth)
        if result:
            return result
    return NULL

function DLS (root, goal, depth)
    if depth == 0:
        if root == goal:
            return root
        return NULL
    for child in root.children:
        result = DLS (root, goal, depth-1)
        if result:
            return result
    return NULL
```



d = 0 :

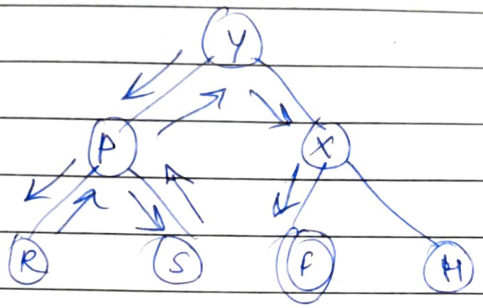
return NULL

d - 1 :



return NULL

d = 2 :



return F //

→ A* on 8 puzzle

pseudocode :

```
function A_star(start, goal):
    open ← priority_queue()
    open.add(start, manhattan(start, goal))
    close = set()
    g_score = dictionary()
    g_score[start] = manhattan(start, goal)
    parent = dictionary()

    while !open.empty():
        curr = open.pop()
        if curr == goal:
            return path(parent, curr)
        close.add(curr)
        for neighbour in neighbours(curr):
            if neighbour in close:
                continue
```

```
        newg = g_score[curr] + 1
        if neighbour not in open and closed or
            newg < g_score[neighbour]:
            parent[neighbour] = curr
            g_score[neighbour] = newg
            fscore = g_score[neighbour]
                    + manhattan(neighbour, go
            open.add(neighbour, fscore)
    return "No soln"

function Neighbours(curr):
    neighbours = []
    if zero can move up:
        neighbours.append(upstate)
    .
    .
    .
    if zero can move right:
        neighbours.append(rightstate)
    return neighbours

function path(parent, curr)
    followed = [curr]
    while curr in parent:
        curr = parent[curr]
        followed.append(curr)
    print(reverse(path))

function manhattan(curr, goal):
    return manhattan distance of each
        element in curr and goal
```

Initial state :

1	2	3
8	0	4
7	6	5

j = man

1	2	3
8	4	0
7	6	5

j = 1 + man

1	2	3
0	8	4
7	6	5

j = 1 + man

1	2	3
8	6	4
7	0	5

j = 1 + man

1	0	3
8	2	4
7	6	5

j = 1 + man

15/10/24