Lab-3

1/09/25

Using BFS solve 8 puzzle without heuristic

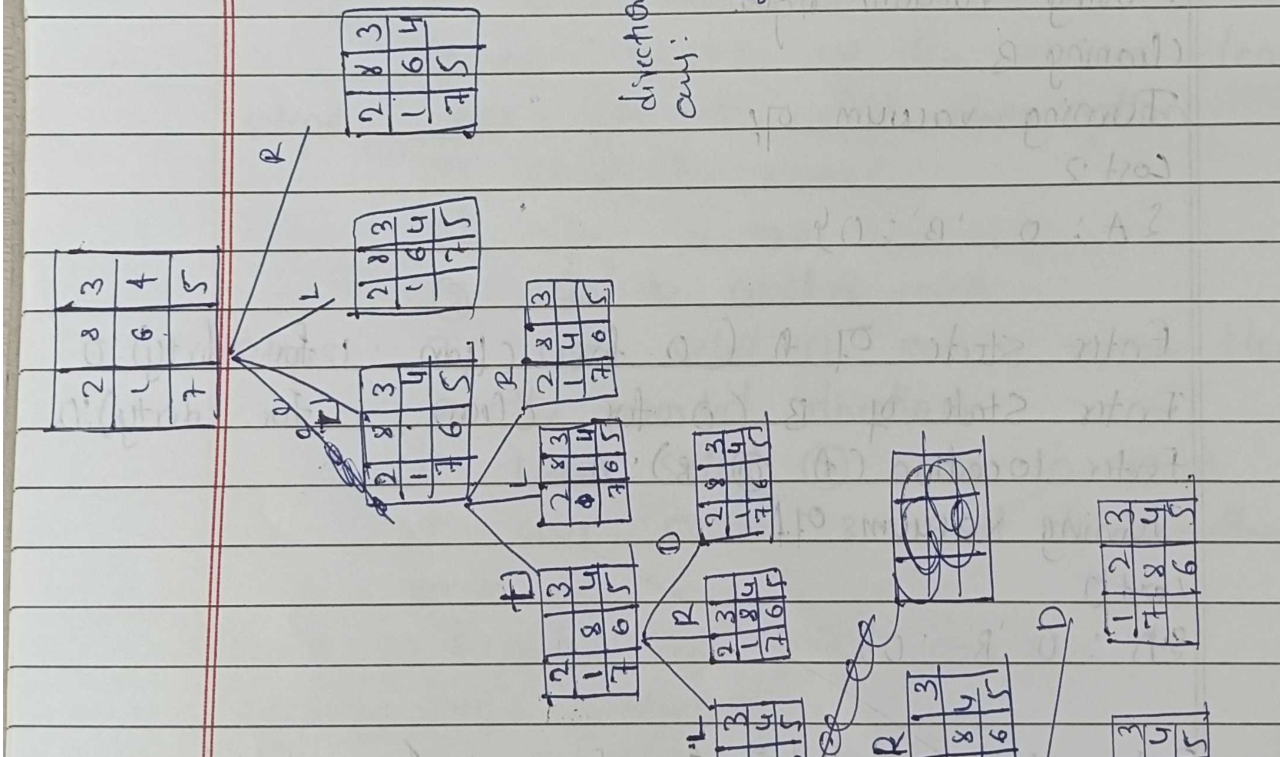
| | | |
|---|---|---|
| 2 | 8 | 3 |
| 1 | 6 | 4 |
| 7 | | 5 |

initial

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 8 | | 4 |
| 7 | 6 | 5 |

goal

direction
ans: Up, Up, Left, Down, Right.



Algorithm

BFS

①. Start. with initial puzzel

②. Check all states you can reach goal in 1 move

③ If goal is not found, then check for the States you can reach goal in 2 ⓞⓡ 3 move. (up, down, rights left).

④. Stops when goal is reached.

DFS

①. Start

②. Pick one move & keep going deeper & until you can't go further.

③. If stuck, go back & try another path

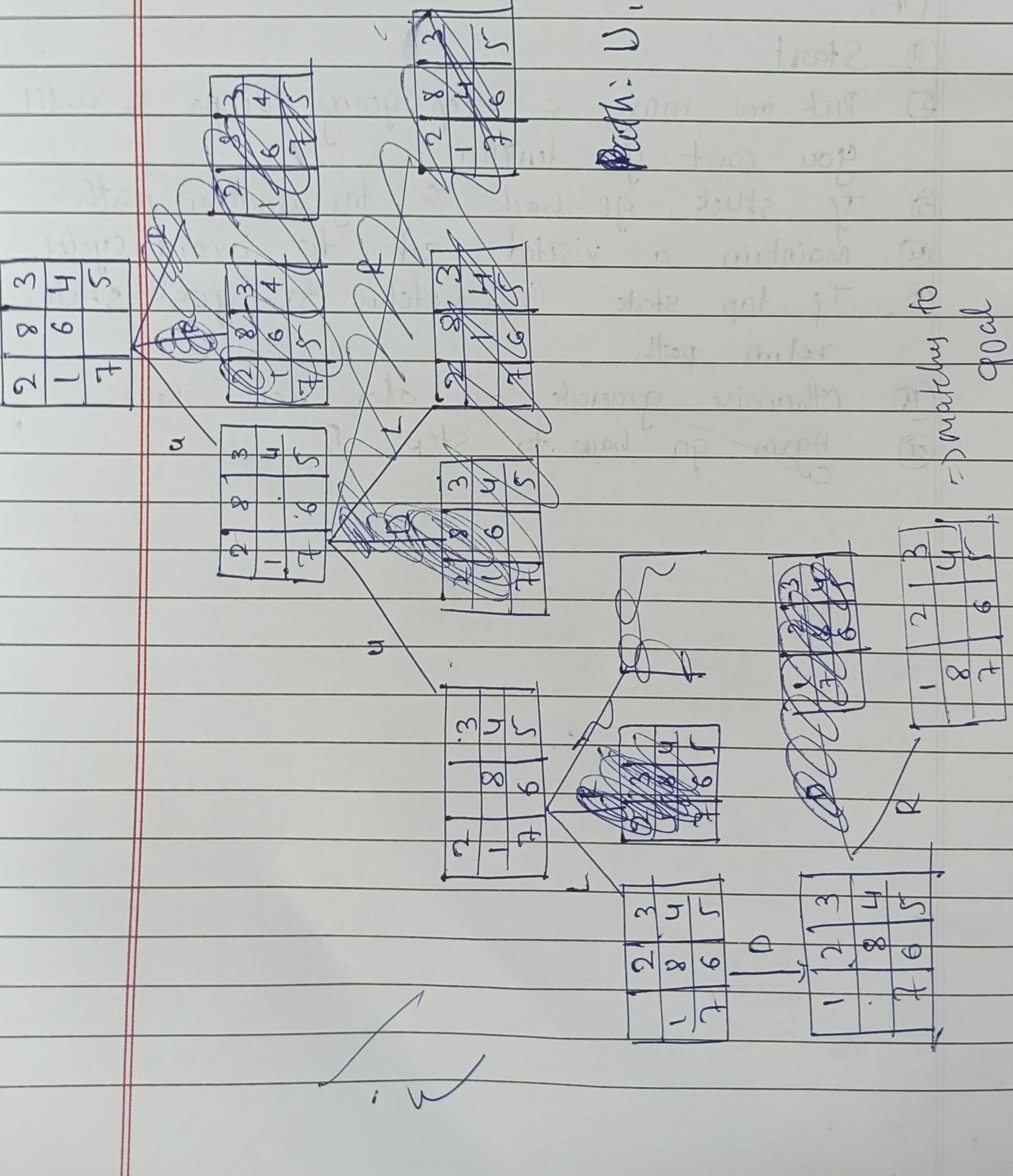
④. Maintain a visited set to avoid cycles.

⑤. If top state is matches to goal state, return path

⑥ Otherwise generate all other next states

⑦. ~~Again~~ go back to step 5.

Same problem using DFS.



Path: U, U, L, D, R

=> matches to goal

O/P.

Using BFS.

2, 8, 3
1.6, 4
7, 0, 5
↓
2, 8, 3
1, 0, 4
7, 6, 5
↓
2, 0, 3
1, 8, 4
7, 6, 5
↓
0, 2, 3
1, 8, 4
7, 6, 5
↓
1, 2, 3
0, 8, 4
7, 6, 5
↓
1, 2, 3
8, 0, 4
7, 6, 5.

Using DFS.

2, 8, 3
1, 6, 4
7, 0, 5
↓
2, 8, 3
1, 0, 4
7, 6, 5
↓
2, 0, 3
1, 8, 4
7, 6, 5
↓
0, 2, 3
1, 8, 4
7, 6, 5
↓
1, 2, 3
0, 8, 4
7, 5, 6
↓
1, 2, 3
8, 0, 4
7, 6, 5.

Iterative Deepening Search (IDS) or Iternative Deepening Depth First Search (IDDFS)

Algorithm

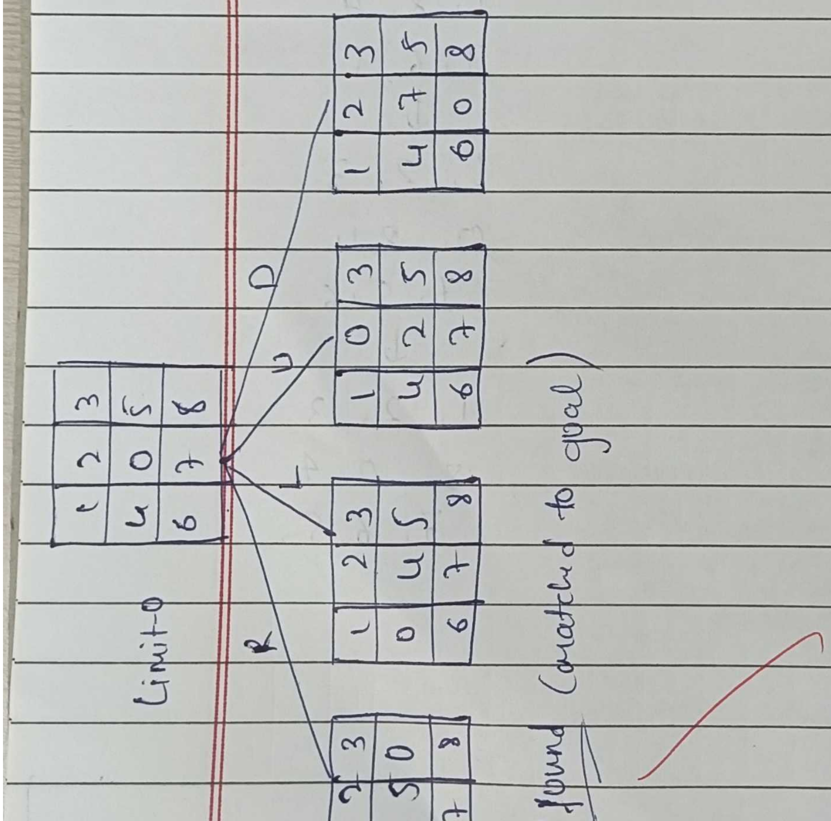
```
function Itr (problem) returns a solution
    inputs: problem, a problem
    for depth ← 0 to ∞ do
        result ← Depth-Limited-Search(problem, depth)
        if result ≠ cuttoff then return result
end
```

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 0 | 5 |
| 6 | 7 | 8 |

initial state

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 0 |
| 6 | 7 | 8 |

Goal.



O/P.

```
searching with depth limit = 0
searching with dept limit =1
Solution found in 1 move
Step 0:
1 2 3
4 0 5
6 7 8

step1:
1 2 3
4 5 0
6 7 8 .
```