

15/10/24

LAB-3

Iterative deepening depth search.

Algorithm

```
bool IDDFS (src, target, max-depth)
```

```
for l from 0 to max
```

```
    if DFS (src, target, limit) == true  
        return true
```

```
return false
```

```
bool DFS (src, target, l)
```

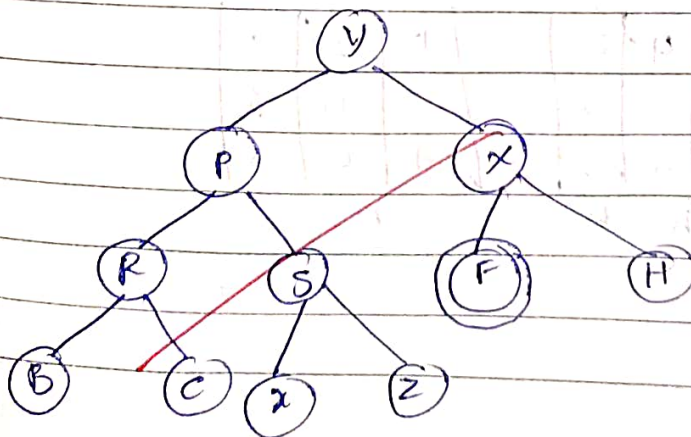
```
    if (src == target)  
        return true;
```

```
    if (l <= 0)  
        return false;
```

```
    foreach adjacent i of src  
        if DFS (i, target, l-1)  
            return true
```

```
return false;
```

$(Y, F, 2)$   
 $\swarrow \searrow$   
 $DLS(Y, F, 0)$       $D$   
 $\downarrow$   
 False.  
 $\downarrow$   
 $DLS(Y, F, 1)$   
 $\downarrow$   
 $DLS(P, F, 0)$   
 $\downarrow$   
 False.  
 $DLS(X, F, 0)$   
 $\downarrow$   
 False.  
 $\downarrow$   
~~DLS~~  $DLS(Y, F, 2)$   
 $\swarrow \searrow$   
 $DLS(P, F, 1)$       $DLS(X, F, 1)$   
 $\swarrow \searrow$       $\downarrow$   
 $DLS(P, F, 0)$       $DLS(F, F, 0)$   
 $\swarrow \searrow$       $\downarrow$   
 false.      $DLS(S, F, 0)$      return true.  
 $\downarrow$   
 false.



# 8 Puzzle using A\*

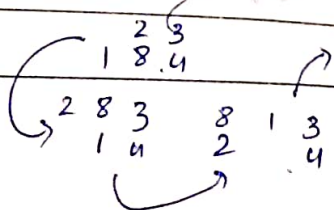
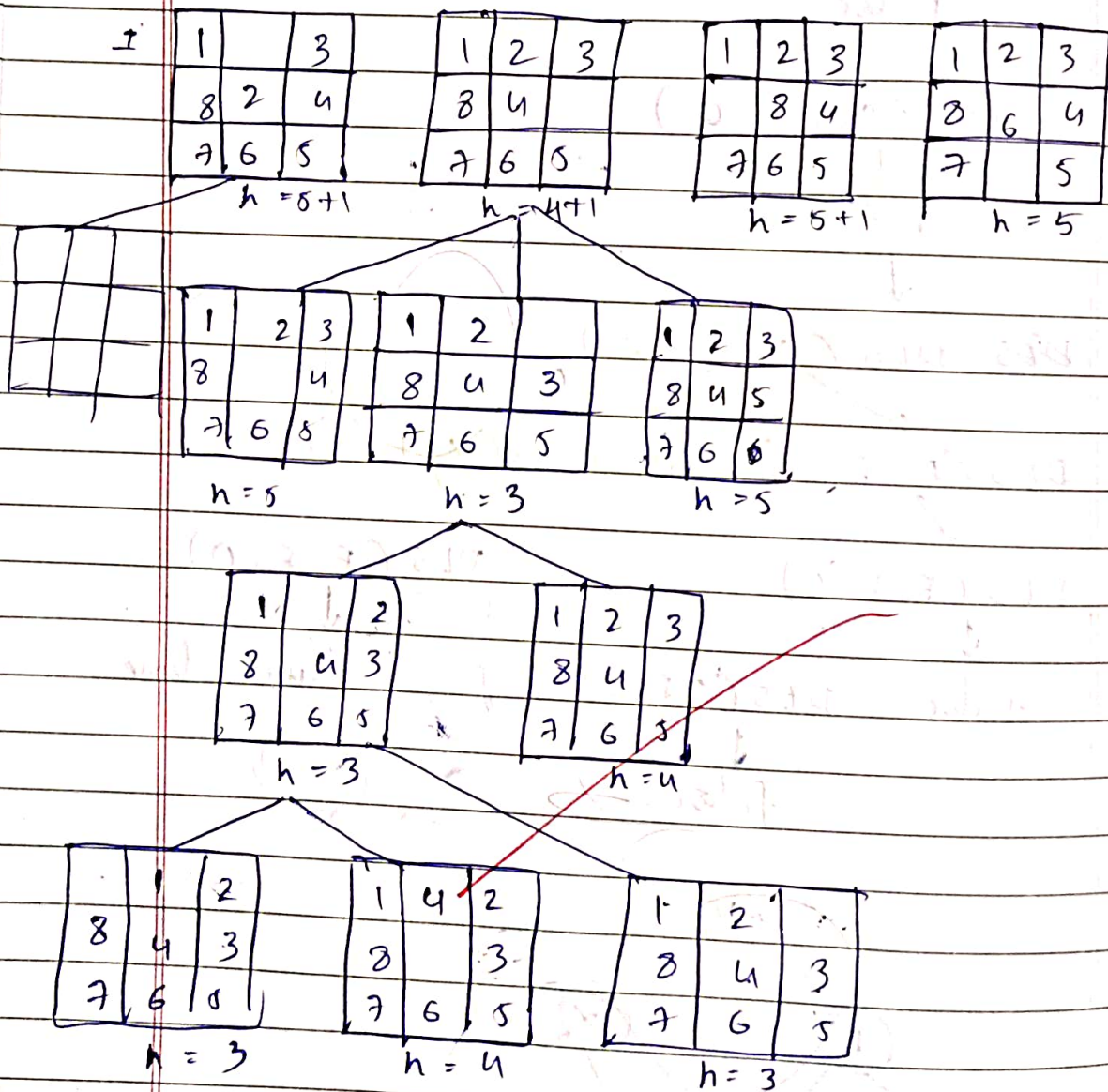
Initial state

1	2	3
8		4
7	6	5

goal state

2	8	1
	4	3
7	6	5

$$g = 0 \quad h = 5$$



481  
32  
765

81  
432  
765

812  
43  
765

81  
432  
765

851  
42  
765



Algorithm:

puzzle().

Step 1: start  
Step 2: initialize startstate and goalstate  
Step 3:

heuristic (state)

{ find no. of mismatched entries;

return value.

Step 4: Find the heuristic values of all the states in the same level.

$\min(g+n)$  for each state.

Step 5: Change current state to the state which has minimum of  $g+n$

Step 6: check if the current state == goal state  
return true

else call the puzzle() iteratively

~~Done~~  
10/10/24