

8-12-23

8-puzzle problem using ID-DFS:

16

Code:

```
def id-dfs(puzzle, goal, get-moves):
    import itertools
```

```
# get-moves → possible-moves
```

```
def dfs(route, depth):
```

```
    if depth == 0:
```

```
        return
```

```
    if route[-1] == goal:
```

```
        return route
```

```
    for move in get-moves(route[-1]):
```

```
        if move not in route:
```

```
            next-route = dfs(route + [move], depth + 1)
```

```
            if next-route:
```

```
                return next-route
```

```
for depth in itertools.count():
```

```
    route = dfs([puzzle], depth)
```

```
    if route:
```

```
        return route
```

```
def possible-moves(state):
```

```
    b = state.index(0)
```

```
    d = []
```

```
    if b not in [0, 1, 2]:
```

```
        d.append('u')
```

```
    if b not in [6, 7, 8]:
```

```
        d.append('d')
```

```
    if b not in [0, 3, 6]:
```

```
        d.append('l')
```

```
    if b not in [2, 5, 8]:
```

```
        d.append('r')
```

```
pos-moves = []
```

```
for i in d:
```

```
    pos-moves.append(generate(state, i, b))
```

```
return pos-moves
```

```
def generate(state, m, b):
```

```
    temp = state.copy()
```

```
    if m == 'd':
```

```
        temp[b+3], temp[b] = temp[b], temp[b+3]
```

```
    if m == 'u':
```

```
        temp[b-3], temp[b] = temp[b], temp[b-3]
```

```
    if m == 'l':
```

```
        temp[b-1], temp[b] = temp[b], temp[b-1]
```

```
    if m == 'r':
```

```
        temp[b+1], temp[b] = temp[b], temp[b+1]
```

```
    return temp
```

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

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

```
    route = id-dfs(initial, goal, possible_moves)
```

```
    if route:
```

```
        print("Success!!")
```

```
        print("Path:", route)
```

```
    else:
```

```
        print("Failed to find a solution")
```

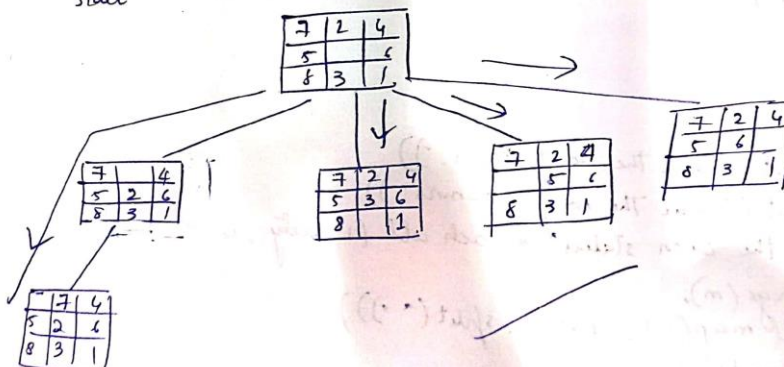
ID-DFS - Combination of BFS and DFS  
- DFS in BFS manner

7	2	4
5		6
8	3	1

Start

	1	2
3	4	5
6	7	8

goal



```
PS C:\Users\neha2\OneDrive\Documents\NehaKamath_11PM21CS113_At1ab> python -u "c:\Users\neha2\OneDrive\Documents\NehaKamath_11PM21CS113_
Success!! It is possible to solve 8 Puzzle problem
Path: [[1, 2, 3, 0, 4, 6, 7, 5, 8], [1, 2, 3, 4, 0, 6, 7, 5, 8], [1, 2, 3, 4, 5, 6, 7, 0, 8], [1, 2, 3, 4, 5, 6, 7, 8, 0]]
```

