

```
def dfs (src, target, limit, visited - state):
    if src == target:
        return True
```

```
    if limit <= 0:
        return False
```

```
    visited - states - append (src)
    adj = possible - moves (src, visited - state)
```

```
    for move in adj:
```

```
        if:
```

```
            dfs (move, target, limit - 1, visited - state):
                return True
```

```
    return False
```

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

```
    ind = state - index (-1)
```

```
    d = []
```

```
    if ind + 3 in range (9):
        d - append ('d')
```

```
    if ind - 3 in range (9):
        d - append ('u')
```

```
    if ind not in [0, 3, 6]:
        d - append ('l')
```

```
    if ind not in [2, 5, 8]:
        d - append ('r')
```

```
    pos - moves - append (gen (state, move, ind))
    return pos - moves - append (more info more if
    in visited - state)
```

```
def gen (State, m, b):
```

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

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

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

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

```
        temp[b] = a
```

```
    elif m == 'u':
```

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

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

```
        temp[b] = a
```

```
    elif m == 'r':
```

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

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

```
        temp[b] = a
```

```
    return temp
```

```
def iddfs (src, target, depth):
```

```
    visited = set()
```

```
    for i in range (1, depth+1):
```

```
        if dfs (src, target, i, visited + set()):
```

```
            return True
```

```
    return False
```

```
# Ex 1
```

```
src = [1, 2, 3, -1, 4, 5, 6, 7, 8]
```

```
target = [1, 4, 3, 6, 5, 8, -1, 7, 2]
```

```
depth = 4
```

```
iddfs (src, target, depth) # returns depth 4
```