

TMItalk: Too Much Information Talk

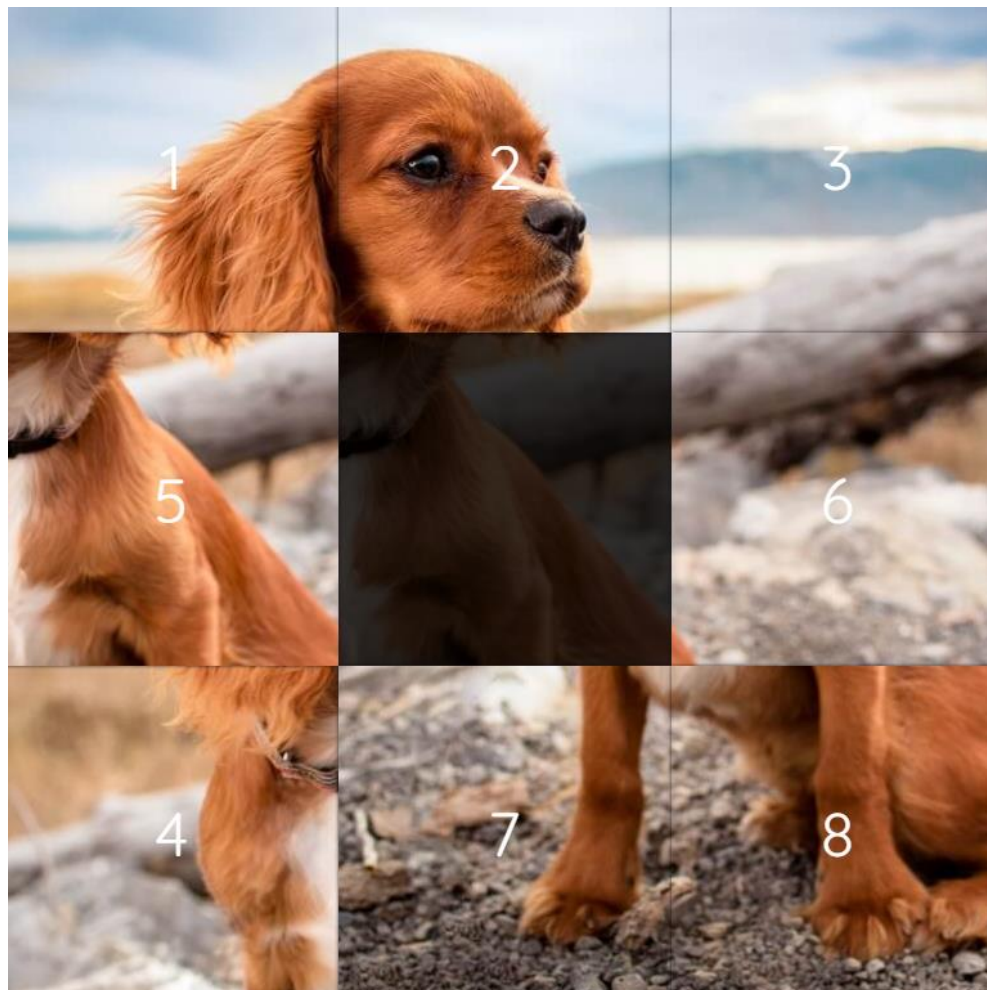
8-퍼즐 문제와 A^* 알고리즘

1. 8-퍼즐 문제를
이해해 보자.





■ 8-퍼즐 문제와 A* 알고리즘:





■ 8-퍼즐 문제:

8	7	6
5	0	4
3	2	1

초기 상태



1	2	3
4	5	6
7	8	0

목표 상태



■ 8-퍼즐 문제의 해결: 보드 상태의 이동

현재 상태

8	7	6
5	0	4
3	2	1

다음 상태

8	0	6
5	7	4
3	2	1

8	7	6
0	5	4
3	2	1

8	7	6
5	2	4
3	0	1

8	7	6
5	4	0
3	2	1



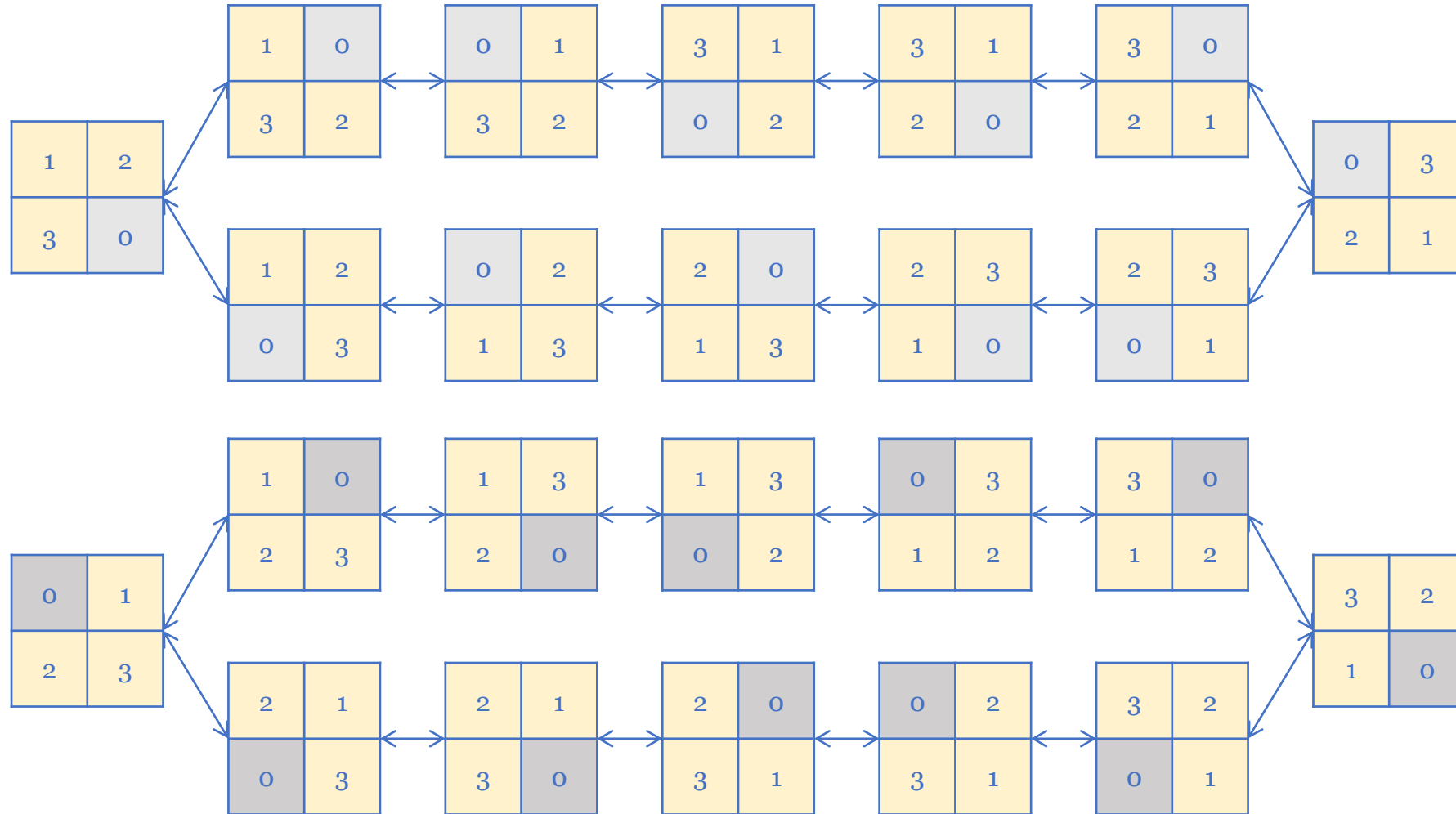
■ 8-퍼즐 문제: 상태 공간

- 8-퍼즐에서 가능한 상태의 수는?
 - 각 칸에 들어갈 수 있는 수가 9개: $9! = 362,880$
- N-퍼즐에서 가능한 상태의 수는?
 - 2×2 보드: $4! = 24$
 - 3×3 보드: $9! = 362,880$
 - 4×4 보드: $16! = 20,922,789,888,000$
- 각 상태에서 가능한 다음 상태의 수는?
 - 모서리에 빈 칸(0)이 있는 경우: 2개
 - 변두리에 빈 칸(0)이 있는 경우: 3개
 - 가운데에 빈 칸(0)이 있는 경우: 4개



TMItalk: 8-퍼즐 문제와 A* 알고리즘

- 중요한 질문: 모든 상태들은 서로 이동 가능한가?
 - 3-퍼즐(2×2 보드)의 경우:





- 목표 상태로 이동 가능함을 판단할 수 있는가?
 - 다음과 같은 역전 카운트로 이동 가능성을 판단 가능하다고 함:
 - SOLVABILITY-CHECK:
 - $n \times n$ 보드에서 빈 칸(o)이 아닌 숫자들을 일렬로 나열했을 때
 - n 이 홀수인 경우, 역전 카운트가 짝수이면 이동 가능
 - n 이 짝수인 경우,
 - 빈 칸의 행(row) 위치가 아래서부터 짝수인 행에 있으면
 - 역전 카운트가 홀수일 때 이동 가능
 - 빈 칸의 행(row) 위치가 아래서부터 홀수인 행에 있으면
 - 역전 카운트가 짝수일 때 이동 가능



```
def is_solvable(n, board):  
    if n % 2 == 1:  
        return count_inversions(board) % 2 == 0  
    else:  
        row_blank = board.index(0) // n  
        if row_blank % 2 == 0:  
            return count_inversions(board) % 2 == 1  
        else:  
            return count_inversions(board) % 2 == 0
```





TMItalk: 8-퍼즐 문제와 A* 알고리즘

■ 왜 그런가?

1	2	3
4	5	6
7	8	0

1 2 3 4 5 6 7 8: 역전 카운트 0

1	2	3
4	5	6
8	7	0

1 2 3 4 5 6 8 7: 역전 카운트 1



```
def count_inversions(board):  
    count = 0  
    for i in range(len(board)):  
        for j in range(i + 1, len(board)):  
            if board[i] != 0 and board[j] != 0 and board[i] > board[j]:  
                count += 1  
    return count
```





■ 전체 상태 공간에서 해결 가능한 상태 확인:

```
N = int(input())
n = int(sqrt(N))
board = []
for i in range(n):
    board += list(map(int, input().split()))
solved = [i for i in range(1, N)] + [0]

count = 0
for perm in permutations(solved):
    if is_solvable(n, perm):
        # print(perm)
        count += 1
print(count, factorial(N) // 2)
```

puzzle.3.1.in:

```
4
0 1
2 3
```

puzzle.8.1.in:

```
9
1 2 3
4 5 6
8 7 0
```

Any Questions?



주니온TV@Youtube

자세히 보면 유익한 코딩 채널