**A\* 알고리즘**

**#개요**

- 경로 탐색 알고리즘

- 최단 경로 탐색에 사용

**#구성**

1. **휴리스틱 함수(Heuristic Function)**

* 기능: 현재 노드에서 목표 노드까지의 예상 비용을 추정
* h(n)으로 표기(n : 현재 노드)
* 실제 비용을 과대평가하는 것에 대해 주의(최적성 보장)

1. **비용 함수(Cost Function)**

* f(n) = g(n) + h(n)
* g(n): 시작 노드에서 현재 노드 ‘n’까지의 실제 경로 비용
* h(n): 현재 노드에서 목표 노드까지의 비용(휴리스틱 비용)

1. **우선순위 큐(Priority Queue)**

* 탐색할 노드 관리
* f(n) 값에 따라 노드를 정렬하여, 가장 적은 비용을 가지는 노드 탐색

**#알고리즘 동작 과정**

1. **초기화**

* 시작 노드를 초기화하고 ‘g’ 값을 0으로 설정
* 시작 노드의 ‘f’ 값을 계산하여 우선순위 큐에 삽입

1. **노드 탐색**

* 우선순위 큐에서 ‘f’ 값이 가장 작은 노드를 꺼냄
* if(꺼낸 노드 == 목표 노드){

탐색 종료 및 경로 반환

}

* else{

현재 노드와 이웃한 노드를 탐색

}

1. **이웃 노드 업데이트**

* 각 이웃 노드에 대해 ‘g’ 값을 계산
* If(이웃 노드가 이전에 탐색되지 않았거나, 새로운 ‘g’ 값이 기존 ‘g’ 값 보다 작은 경우, 이웃 노드의 ‘f’ 값을 갱신하고 우선 순위 큐에 삽입

1. **반복**

* 목표 노드를 찾거나, 우선 순위 큐가 비워질 때까지 위 과정을 반복함

텍스트, 사람, 의류, 인간의 얼굴이(가) 표시된 사진

자동 생성된 설명

1. Mismatched Count

R = 3{4, 5, 8이 Goal과 다른 상황}

1. 빈칸이 움직일 수 있는 부분에 대해서 계산

위 상황에서는 상, 우, 하 3방향으로 이동 가능한 상황

텍스트, 사람, 스크린샷, 의류이(가) 표시된 사진

자동 생성된 설명

UP. 상 방향 이동 시

스케치, 텍스트, 친필, 그림이(가) 표시된 사진

자동 생성된 설명

DOWN. 하 방향 이동 시

텍스트, 친필, 스케치, 폰트이(가) 표시된 사진

자동 생성된 설명

RGIHT. 우 방향 이동 시

텍스트, 친필, 스케치이(가) 표시된 사진

자동 생성된 설명

휴리스틱을 사용하면 가장 최고의 값이 어떤 것인지 알 수 있다.

휴리스틱 값을 계산 해보겠다.

[UP. 상 방향 이동 시]: misplaced된 값의 개수: 4(공백은 세지 않는다)

스케치, 텍스트, 친필, 그림이(가) 표시된 사진

자동 생성된 설명

[DOWN. 하 방향 이동 시]: misplaced된 값의 개수: 4

텍스트, 친필, 스케치, 폰트이(가) 표시된 사진

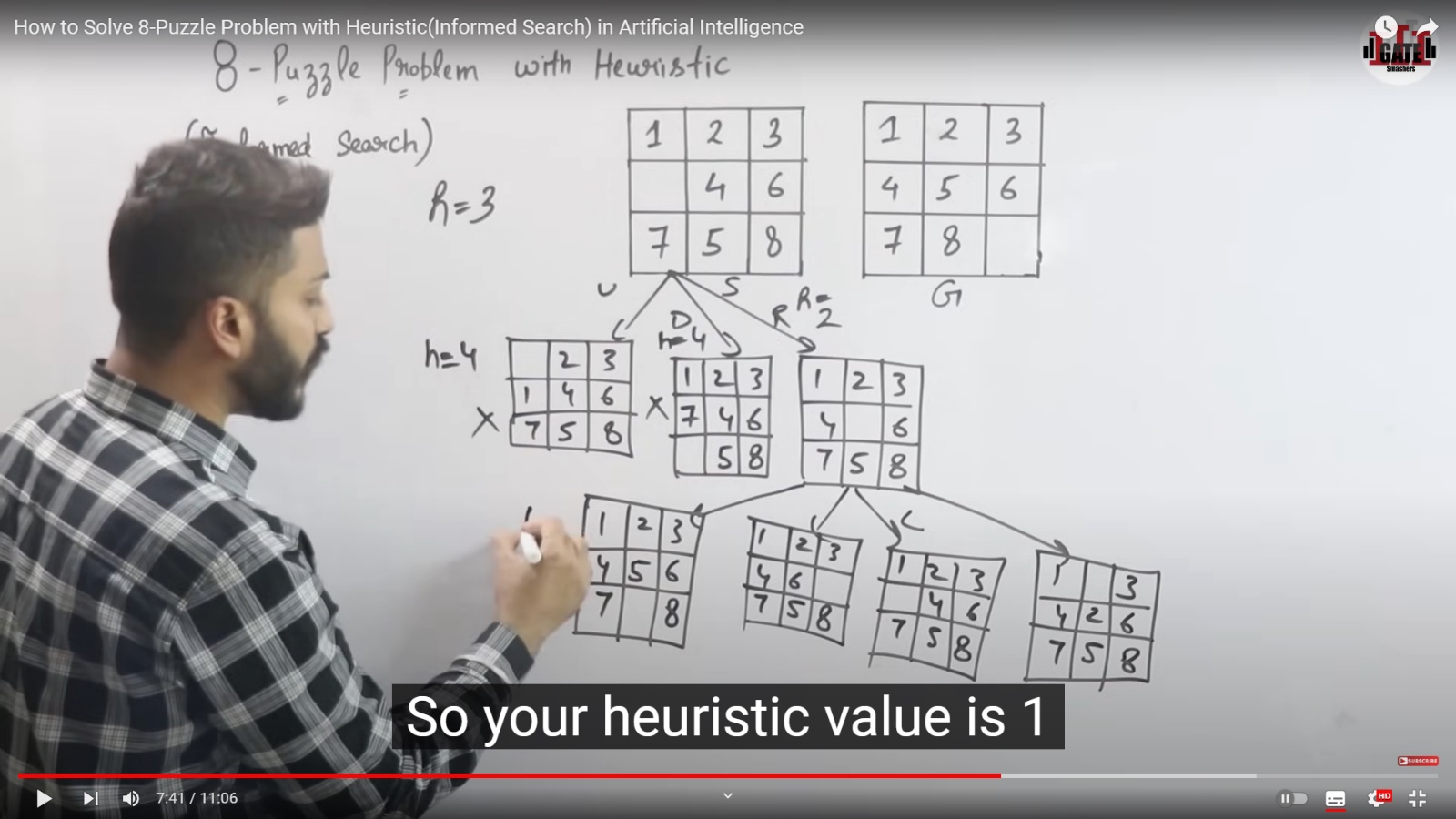
자동 생성된 설명

[RGIHT. 우 방향 이동 시]: misplaced된 값의 개수: 2[가장 작은 값] – 휴리스틱 결과[2]

휴리스틱이 작다는 것은 가장 빠른 길이라는 의미이다.

텍스트, 친필, 스케치이(가) 표시된 사진

자동 생성된 설명



1. 하방향 1[이걸로 간다]

텍스트, 친필, 화이트보드, 낱말맞추기 퍼즐이(가) 표시된 사진

자동 생성된 설명

1. 우방향 3

친필, 스케치, 텍스트, 그림이(가) 표시된 사진

자동 생성된 설명

1. 좌방향 3

텍스트, 친필, 스케치, 그림이(가) 표시된 사진

자동 생성된 설명

1. 상방향 3

텍스트, 친필, 스케치, 화이트보드이(가) 표시된 사진

자동 생성된 설명

텍스트, 의류, 사람, 스크린샷이(가) 표시된 사진

자동 생성된 설명