풀이 방법

Dp를 사용하는 문제.

틀린 풀이 :

재귀를 사용해서 왼쪽->오른쪽으로 이동하면서 첫 시작 문자와 맞는 발판이 있다면 재귀함수를 통해 모든 가짓수를 계속 구했다. 때문에 동일한 계산을 자주 반복하여(동일한 발판, 동일한 글자인 경우 뒤의 계산은 어차피 동일) 시간초과가 나왔다고 판단한다.

우선 동일한 시작 발판을 찾아 재귀함수를 불렀다면 해당 발판 오른쪽부터 다리 끝까지 검색하면서 반대 발판(천사 or 악마)에 찾고자 하는 문자열의 다음 문자가 나오는 인덱스를 찾아 다시 재귀함수를 호출하여 결과값을 전달 받고 모든 결과값을 누적하여 값을 반환하는 방식을 사용했다. Dp[][]는 dp[발판 종류][발판 위치]를 사용하여 구했다. (메모이제이션 할 필요가 없었으므로 2차원 배열로도 충분했다.)

정답 :

재귀를 사용하는 건 같지만, 메모이제이션을 통해서 dp값을 저장하고 차후에 동일한 패턴의 값을 검색할 경우 값을 가져오는 방식으로 답을 구했다.

여기서 위와는 다르게 각 발판에 도달했을 때 구해야 하는 글자가 다를 수 있다.(똑같은 발판이라도 RDR이라는 문자열에서 0인덱스 R인지 2인덱스 R인지를 구별해야한다.) 따라서 메모이제이션을 사용할 경우에는 DP[발판 종류][발판 위치][구할 문자열 인덱스]라는 3차원 배열을 사용하여 구했다.

DP는 -1로 시작 전 초기화를 진행하고, 값을 구하는 과정은 위와 동일하다. 이 때, 재귀함수 초반부에 DP의 변동유무(-1이 아닌 경우)를 잡아서 바로 DP값을 반환해주고 아래 계산은 패스시킨다. 이렇게 함으로서 한 번 계산해 둔 값을 다시 계산할 필요가 없어지게 된다.

