알고리즘 과제 보고서

* HW4

2013147550 김지호

2013147544 조인

1. 전제 조건 분석

전체 game board를 규칙에 맞게 돌을 꽉 채워서 올릴 경우 들어가는 돌의 갯수는 2N개이다. 따라서, 2N개의 흰 돌과 2N개의 검은 돌이 주어져 있으므로 흰 돌 또는 검은 돌 한 종류만으로도 전체 game board를 꽉 채울 수 있다. 그러므로 주어진 game board의 모든 음수값들을 양수로 바꾼 후, 돌의 종류가 흰 돌 하나만 있다고 문제를 변형하여 진행한다.

1. 문세 세분화

1개의 column들을 기준으로 (세로 한 줄) 문제를 생각한다. 하나의 column에서 돌을 놓을 수 있는 경우의 수는 8가지이다. 또한, game board에 돌을 왼쪽에서부터 오른쪽으로 차례대로 놓는다고 생각할 경우, 특정 column t에 돌이 배치되었을 때 영향을 받는 column은 바로 오른 쪽 column 뿐이다. (왼쪽 column은 이미 놓여져 있을 것이므로). 따라서, 특정 column t에 돌을 놓는 경우의 수는 바로 왼쪽의 column에 영향을 받고, column t에 돌이 놓여지면 이는 바로 오른쪽 column에 영향을 줄 것이다. 위와 같은 사실들을 바탕으로 본 문제를 Dynamic Programming을 활용하여 답을 구한다.

1. DP 구현

game board의 column을 왼쪽부터 0번째, 1번째, N-1번째 column이라 부르자. t번째 column부터 시작하는 game board의 일부분 (substructure)에서 돌을 놓아 구할 수 있는 최대값은, t번째 column에서 돌을 배치할 수 있는 case 마다, t번째 column에서 나오는 숫자들의 합과 t+1번째 column부터 시작하는 game board의 일부분에서 돌을 놓아 구할 수 있는 최대값을 구한 후 이 중 최대가 되는 case의 값이 t번째 column부터 시작하는 game board의 최대값이다. 이를 Pseudo Code로 표현해 보면 다음과 같다.

|  |
| --- |
| function maxValue(t) {  max = -1;  for available\_cases {  value = sumOfColumn(t) + maxValue(t + 1);  if (value > max) max = value;  }  return max;  } |

전체 N개의 column을 가진 game board를 iteration을 진행하면서 N-1개, N-2개, …, 1개의 column을 가진 game board를 subproblem으로 설정하고, 이 subproblem의 최적해를 이용하여 본 문제, N개의 column을 가진 game board의 최댓값을 구하는 Dynamic Programming 방식으로 본 문제의 최적해를 구한다.