10주차 예비보고서

전공: 컴퓨터공학과 학년: 4학년 학번: 20212022 이름: 이예준

**1.**

이번 실습에서 구현하는 추천 기능은 블럭의 회전, 위치에 따른 모든 경우의 수의 점수와

field 정보들을 바탕으로 Tree구조로 생성한다. 처음에 root Node에서 현재 블럭의 회전, 위치에

따라 얻는 누적점수, field에 대한 모든 경우의 정보들을 child Node들에 저장한다.

또 각 child Node들은 다음 블럭의 회전, 위치에 따라 얻는 누적점수, field에 대한 모든 경우의

정보들을 child Node의 child Node들에 저장한다. Tree의 Depth는 고려해야 하는 블럭의 개수에

따라 결정된다( 고려해야 하는 블럭 3개 = Tree Depth는 3 ). 누적 점수 중 최고점수가 되는

경우에 대한 정보를 따로 저장하여 현재 블럭이 어느 회전과 위치로 가야 하는지 추천한다.

Tree 구조를 이용하면 모든 경우의 수에 대해 고려를 하기 때문에 최고점을 만드는 가장 좋은

결정을 할 수 있도록 도와준다는 장점이 있지만 모든 경우의 수를 구하는 특성에 의해

고려해야 하는 블럭의 개수가 많으면 많을수록 계산량이 기하급수적으로 많아지고,

재귀를 통해 계산을 하기 때문에 메모리 Overflow 발생 가능성도 존재한다.

**2.**

가자치기(pruning)을 이용하여 계산 중간에 최대 점수를 넘지 못하는 경우에 대해서는

해당 경로에 대한 계산을 더 이상 진행하지 않는다. 이를 통해 계산 시간과 메모리 사용을

충분히 줄일 수 있다. 하지만 이런 방식으로 구현하게 될 경우 greedy 특성에 따라 최적해를

구하지 못할 수도 있다. 따라서 완전 탐색과 가지치기 사이 중간에 있는 해결책으로

빔 탐색(Beam Search)를 이용해 볼 수 있을 것이다. 빔 탐색은 Tree의 각 Depth마다 최고점을

달성할 것 같은 후보들을 n개 선택해서 그 후보들에 대한 경로에 대해서만 계산을 진행한 뒤

최종적으로 가장 큰 점수를 가진 후보를 선택한다.