전공 : 컴퓨터공학과 학년 : 4 학번 : 20212020 이름 : 박민준

**1. 교재를 참조하여 테트리스 프로젝트 3주차에 구현하는 추천 기능은 어떤 원리로 작동되는지 설명하시오. 그리고 추천 기능을 구현하는 tree 구조의 장점(효율성)과 단점(비효율성)을 기술하시오.**

- 테트리스 3주차 프로젝트에서 구현하는 추천 시스템은 사용자가 더 많은 점수를 얻기 위해 블록을 놓아야 할 최적의 위치를 추천하는 기능이다. 이 기능은 현재 블록과 다음 두 개의 블록을 고려하여 다양한 블록 배치 시나리오를 평가하고, 가장 높은 점수를 얻을 수 있는 블록의 위치를 추천한다. 추천 과정은 다음과 같은 단계로 구성된다.

1) Tree 구조 생성

- 현재 블록과 다음 두 개의 블록에 대해 가능한 모든 배치 시나리오를 트리 구조로 나타낸다. 트리의 각 노드는 블록의 ID, 위치, 회전 수, 필드 상태, 누적 점수 등의 정보를 포함한다.

2) 점수 계산

- 각 배치 시나리오에 대해 누적 점수를 계산한다. 누적 점수는 블록이 놓인 후의 필드 상태와 삭제된 라인 수 등에 기반한다.

3) 최적 경로 탐색

- 트리의 루트 노드부터 시작하여 가장 높은 누적 점수를 갖는 경로를 찾는다.

4) 추천 위치 표시

- 최적 경로에 해당하는 블록의 위치와 회전 수를 사용자에게 추천한다. 추천된 블록의 위치는 'R'로 표시된다.

- Tree 구조의 장점과 단점은 다음과 같다.

1) 장점

- Tree 구조를 통해 여러 단계에 걸친 블록 배치를 시뮬레이션 할 수 있으므로 단순히 현재 블록만이 아닌 다음 블록까지 고려하여 최적의 위치를 추천할 수 있다. 또한, 다양한 배치 시나리오를 통해 가장 높은 점수를 얻을 수 있는 경로를 찾을 수 있다.

2) 단점

- 시간 복잡도 측면에서 트리의 깊이가 깊어질수록 고려해야 할 노드의 수가 기하급수적으로 증가하여 계산 시간이 매우 오래 걸릴 수 있다. 예를 들어, 각 블록에 대해 34가지 배치를 고려할 때, 3개의 블록을 고려하면 최대 39,304개의 노드를 탐색해야 힌다.

- 또한, 공간 복잡도 측면에서 각 노드는 많은 정보를 포함하고 있으며, 트리의 노드 수가 기하급수적으로 증가함에 따라 메모리 사용량이 크게 증가한다. 예를 들어, 4개의 블록을 고려하면 최대 1,336,336개의 노드를 저장해야 한다.

**2. Tree 구조의 비효율성을 해결할 방법에 대해서 2가지 이상 생각하고, 그 idea를 기술하시오.**

1) Pruning (가지 치기)

- 트리 구성 중에 유망하지 않은 경로를 미리 제거하여 탐색 범위를 줄인다. 예를 들어, 현재까지의 누적 점수가 일정 임계값 이하인 경로는 더 이상 탐색하지 않고 가지를 잘라낸다. 이렇게 함으로써 트리의 노드 수를 줄여 탐색 시간을 단축할 수 있다. 그러나, 가지 치기 과정에서 최적 경로를 놓칠 가능성이 있으므로 주의해야 한다.

2) Data Simplification (데이터 단순화)

- 각 노드에 저장되는 데이터를 단순화하여 메모리 사용량을 줄인다. 예를 들어, 필드 상태를 전체 상태 대신 각 열의 최대 높이로만 저장한다. 이렇게 함으로써 메모리 사용량을 줄일 수 있으며, 노드의 정보를 간단히 하여 탐색 속도를 높일 수 있다. 그러나 데이터 손실로 인해 정확도가 떨어질 수 있으며, 필드의 세부 상태를 반영하지 못할 수 있으므로 주의해야 한다.

- 다음과 같은 두 가지 방법을 적용하여 트리 구조의 비효율성을 개선할 수 있으며, 이러한 개선을 통해 더 효율적이고 정확한 추천 시스템을 구현할 수 있다.