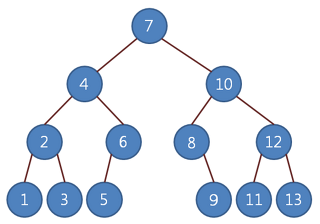
11주차 예비보고서

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.

노드를 사용하는 경우에 tree 구조를 사용한다. 노드를 사용할 경우 모든 노드를 거치지 않고서도 자신의 자리를 찾아갈 수 있다. 이 방법을 이용하면 depth 에 따라 발생할 수 있는 모든 경우의 수를 고려할 수 있다. 이를 고려한 후 이들의 점수들 중 가장 높은 경우를 찾아 이 방식을 추천하도록 한다. 이는 현재 상황에서 점수를 고려했을 때 가장 높은 점수를 낼 수 있는 방법으로 플레이 하는 것으로 점수 측면에서 greedy 한 방법이라고 할 수 있다.



그림과 같이 tree 구조를 이용하면 한 블럭을 놓았을 때 여러가지 경우에 각각에 대하여 다음 블럭을 놓았을 때의 상황을 예측할 수 있다.

하지만 이 방식을 채택할 경우 depth가 조금만 깊어질 경우 시간이 기하급수적으로 증가한다. 또한 한 상황에 대해 점수를 최대로 받을 수 있는 경우를 구하기 때문에 오랫동안 플레이 할 수 있는 수를 두지 못하는 경우가 생길 수 있다. 다시말해 한 블럭을 두었을 때는 가장 높은 점수를 얻을 수 있는 곳에 두지만, 결과적으로는 많은 점수를 내지 못하게 될 수 있다.

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

tree구조를 사용하면 모든 경우의 수를 고려해 최대 점수를 얻는 방식으로 가는 것이다. 블록 면이 바닥에 많이 닿을 수록 점수가 더욱 높아지기 때문에 이 방식을 이용하면 블록이 바닥에 가장 많이 닿게 플레이를 할 가능성이 높다. 하지만 아까 말했듯이 플레이를 함에 있어서 중요한 건 현재의 점수가 아닌 경우가 많다. 현재는 많은 점수를 내지 못한다고 하더라도 오랫동안 진행했을 때 더 오랫동안 플레이를 할 수 있도록 만드는 알고리즘이 필요하다. 또한 tree 구조를 사용한 방식은 앞서 말했듯이 시간이 굉장히 많이 걸린다.

이 때문에 tree 방식이 아닌 다를 방식을 사용할 수 있다. 이를 크게 나누었을 때 첫번째로는 game over가 나지 않도록 높이가 높지 않도록 블럭을 놓는 경우가 있을 수 있다. 이 경우 높이가 최대한 낮아지도록 게임을 플레이하므로 게임이 오래 유지되도록 할 수 있다. 예를 들어 가장 긴 블럭이 들어왔을 경우, 이를 세로로 사용해 최대한 많은 줄을 삭제하도록 하는 행동을 취하도록 할 수 있다. 이는 다른 블럭에 대해서도 적용된다.

두번째 방식으로는 빈 공간이 없도록 하는 방식이 있다. 빈 공간이 없도록 한다는 것은 블럭을 놓았을 때 field 내에 0이 생기지 않도록 하는 방식이다. 이는 delete line을 하기 용이하게 만들기 위해서 중요하다고 할 수 있다.

세번째 방식으로는 field를 평평하게 만드는 것이 있다. field를 평평하게 만들면 한 쪽으로만 블럭이 쏠리면서 죽는 현상을 방지할 수 있다. 빈 공간을 채우게 하는 방식으로 알고리즘을 짤 경우 빈 공간이 없도록 만들기 위해 블럭을 맞추어 놓다가 한 쪽 높이만 높아지면서 죽는 경우가 생길 수 있다. 이를 방지하기 위해서 field가 고르지 않을 경우 높이가 낮은 곳에 놓도록 우선도를 높이는 방식을 채택하는 것도 바람직하다고 할 수 있다.