10주차 예비보고서

전공: 경제학과 학년: 4학년 학번: 20180501 이름: 김연수

1. 추천기능 원리와 tree구조의 장단점.

추천기능은 recommend 함수에서 구현한다. recommend 함수에서 블록들이 어디로 떨어질 지 미리 예측해서 떨어질 블록 중 가장 많은 점수를 낼 수 있는 위치를 선정할 수 있다. 블록 예측은 tree구조로 구현한다. 가령 첫 번째 블록이 rotation = 0, x = 0에서 그대로 아래로 떨어진 경우, 그 필드를 저장하고 다음 블록의 경우를 계산한다. 이를 그림으로 그려보면 tree구조로 표현될 수 있으며, 하나의 경우의 수 아래에 또 여러가지 경우의 수가 붙게 되는 형태인 것이다.

recommend함수의 구현은 다음과 같이 이루어진다. 일단 노드 구조체를 구현해야 한다. 노드의 fileld에 반드시 포함해야 하는 요소는 tree의 level, score, field, child 정보다. 다음 나타날 블록의 모든 rotate, x좌표에 대해서 checkBlock 함수를 호출해 블록이 갈 수 있는 위치인지를 검사한다. 새로운 노드를 하나 생성하고, 정보를 담는다. 정보를 담은 노드를 인자로 넘겨주어 점수를 산출하는 함수인 addBlocktoFiled와 deleteFiled함수를 호출한다. 호출하고나서 얻게 된 field정보와 score를 노드에 업데이트 한다. 이후 current\_level이 MAXLEVEL보다 작은 경우, 즉 마지막 노드에 닿지 않은 경우에 recommend 함수를 재귀 호출해 같은 작업을 반복한다. 만약 위치하고 있는 곳이 MAXLEVEL에 해당한다면, 계산한 score를 maximum score와 비교해 가장 큰 score를 찾아주도록 한다.

해당함수를 구현할 때 tree 구조를 이용한다. tree 구조를 이용해서, rotate와 x좌표의 모든 경우의 수를 탐색 가능하다. 같은 level에서는 같은 블록 id의 정보만이 고려된다. 이런 것들이 알고리즘을 설계하는 데 혼동을 줄여줄 수 있다.

tree 구조를 사용할 때 모든 경우의 수를 편리하게 탐색할 수 있다는 장점이 존재하지만, 모든 경우의 수를 탐색하려고 보면 불필요한 작업이 대거 발생하게 될 수밖에 없다. 한 노드에서 생성하는 child 노드의 수가 34개라고 가정한다면, 블록을 4개까지 고려할 때 생성되는 노드의 수는 34^4 = 1336336개이다. 너무 많은 비용을 소모하게 된다.

1. 비효율성 해결방법
   * 1. pruning

Tree에서 불필요한 branch를 자른다. score가 낮아 가망이 없어 보이는 줄기를 가지치기 한다. 고려되는 노드가 감소하여 시간과 공간이 단축된다는 장점이 있지만, 자른 branch에서 누적 score가 가장 큰 경로가 만들어졌을 지 모르는 일이다.

* + 1. 데이터의 단순화

Node의 정보를 표현하기 위한 더 단순한 방법을 고안해낸다. 현재 가장 많이 소비되고 있는 공간이 field이므로, field의 정보를 2차원 배열이 아닌, 1차원 배열로 바꾼다. field의 높이 만을 기억한다면 가능하다. 하지만 높이만은 기억하게 되면, 중간에 비어있는 공간을 놓치게 되는 현상이 발생할 수 있다.