10주차 예비보고서

전공: 경영학과 학년: 3학년 학번: 20190963 이름: 한다현

**1.**

추천 시스템은 트리 구조를 이용한다. 이번 실험에서는 현재 블록과 다음 두 개의 블록, 총 세 개의 블록을 고려한다. tree의 level1에서는 현재 블록을, level2에서는 다음 블록을, level3에서는 세번째 블록을 고려한다. 먼저 현재 블록의 rotation을 고려하여 현재 블록이 필드에 위치할 수 있는 경우를 모두 찾는다. 각각의 경우에서 다음 블록의 rotation을 고려하여 다음 블록이 필드에 위치할 수 있는 경우를 모두 찾는다. 마지막으로 세 번째 블록의 rotation을 고려하여 세 번째 블록이 필드에 위치할 수 있는 모든 경우를 찾는다. level1에서 level3까지 가능한 경우를 모두 조사하는 과정에서 누적 점수를 함께 계산하는데, 모든 경우를 조사한 후에 level1에 해당하는 노드 중 누적 점수가 가장 높은 경우를 선택하는 알고리즘이 추천 시스템 알고리즘이다. 이렇게 tree 구조를 사용하게 되면, level1에서부터 level3까지, 설계자가 원한다면 더 높은 단계까지 해당하는 모든 경우를 조사하여 각각의 경우의 누적 점수를 비교하여 최선의 결과를 도출할 수 있다는 장점이 있다. 또한 tree 구조를 사용하면 재귀 함수를 통해 누적 점수를 쉽게 계산할 수 있다. 하지만 tree 구조를 사용하면 비효율성 역시 존재한다. 먼저 시간이 굉장히 오래 걸릴 수 있다는 단점이 있다. rotation을 고려하면 한 블록 당 위치할 수 있는 경우가 최대 34가지이다. 이를 3단계까지 진행하면 총 34^3번의 계산이 필요하다. 만약 3단계가 아닌 n단계까지 조사한다면 총 34^n번의 계산이 필요하기 때문에 최적의 결과를 도출하는 데 굉장히 오랜 시간이 걸릴 수 있다는 단점이 있다. 두 번째 문제점은 공간의 비효율성이다. 각각의 노드는 필드의 데이터를 저장해야 한다. 따라서 n단계의 tree를 조사하게 되면 34^n 개의 노드가 각자의 필드를 저장해야 하는데, 필드 데이터의 크기가 c라고 한다면 전체 필드 데이터가 차지하는 메모리 공간은 c\*34^n이다. 메모리 공간 역시 tree의 level이 커질수록 기하급수적으로 늘어날 수 있다는 단점이 있다.

**2.**

시간의 비효율성을 해결하기 위해서는 계산할 경우의 수를 최대한 줄여야 한다. 이 때 사용하는 대표적인 방법인 pruning tree이다. 하지만 이 방법을 사용할 경우 누적 점수를 계산하지 않은 노드에서 최대 누적 점수를 만드는 경로가 존재할 수도 있다. 따라서 이 경우를 해결해야 한다. 누적 점수에 가장 큰 영향을 미치는 것은 줄이 삭제되는 것이다. 블록이 필드에 추가될 때 블록의 밑면이 다른 블록이나 필드와 닿는 경우에는 면적당 10점만 부여되지만 줄이 삭제되는 경우에는 한 줄 당 100점이 부여된다. 따라서 tree의 가지를 지울 때 점수를 고려하는 것이 아니라 삭제된 줄의 개수를 고려한다. 각 노드에서 삭제된 줄의 수가 가장 많은 노드를 위주로 계산을 실행하고, 만약 노드에서 삭제된 줄의 수가 같다면 노드를 삭제하지 않고, 다음 level의 노드를 계산한다.

공간의 비효율성을 해결하기 위해서는 노드에서 저장하는 필드의 데이터를 줄여야 한다. 그 방법 중 하나는 일차원 배열을 사용해 x 좌표에 해당하는 필드의 높이만 저장하는 방법이 있다. 그러나 이 방법을 사용하면 비어있는 부분을 체크하지 못하고 그렇게 되면 줄이 삭제되는 알고리즘이 실행될 때 실제와 다른 결과가 발생할 수 있다. 따라서 x 좌표에 해당하는 필드의 높이와 그 x 좌표의 필드 중 비어있는 y좌표와 그 개수를 저장하는 구조체를 정의하는 방법을 고려해 볼 수 있을 것이다. 그리고 그 구조체를 사용한 일차원 배열을 통해 필드의 정보를 저장하면 필드의 모든 공간을 저장할 때보다 메모리 공간을 아낄 수 있고, 높이만 저장했을 때 발생하는 문제점을 예방할 수 있다.