

전공: 국제한국학

학년: 4

학번: 20181202 이름: 김수미

1. 실습 시간에 작성한 프로그램의 알고리즘과 자료구조를 요약하여 기술하시오. 완성한 알고리즘(추가 구현하게 되는 효율성을 고려한 tree도 포함)의 시간 및 공간 복잡도를 보이시오.

- **알고리즘**: 현재 테트리스 블록과 앞으로 등장하게 될 테트리스 블록에 대한 정보를 바탕으로, 테트리스 블록이 필드에 놓일 수 있는 모든 경우의 수를 고려하여 가장 높은 점수 효율을 보이는 자리로 테트리스 블록이 놓일 자리를 추천한다.

- **자료구조**: tree 자료구조를 사용한다. 현재 블록과 앞으로 등장하게 될 블록들 간의 수많은 경우의 수를 체계적으로 관리할 수 있는 자료구조이다.

- **기존 Tree 시간 및 공간복잡도**: 총 n 개(현재 블록+앞으로 등장하게 될 블록)의 블록을 고려하고 각 블록에 대해서 최대 34가지 위치가 고려된다고 가정하면 시간복잡도와 공간복잡도는 $O(34^n)$ 이다.

- **효율성을 고려한 Tree 시간 및 공간복잡도**

1) **Pruning**: 총 n 개(현재 블록+앞으로 등장하게 될 블록)의 블록을 고려한다고 할 때, 각 블록에 대해 최대 34가지 위치가 고려되었으나 일정 점수보다 낮은 score를 가지는 branch를 자름으로써 약 15가지 정도로 경우의 수를 조정할 수 있다면 시간복잡도와 공간복잡도를 약 $O(15^n)$ 정도로 감소시킬 수 있다.

2) **데이터의 단순화**: 기존에 HEIGHT(22)*WIDTH(10) 개수 만큼의 정보를 저장하기 위한 배열이 필요했다면, 데이터를 단순화 시키는 경우 WIDTH(10) 개수 만큼의 정보를 저장하기 위한 1차원 배열만 있으면 되기 때문에 공간복잡도를 $O(220)$ 에서 $O(10)$ 정도로 감소시킬 수 있다.

2. 모든 경우를 고려하는 tree 구조와 비교해서 어떤 점이 더 향상되고, 어떤 점이 그렇지 않은지 아울러 기술하시오.

1) **Pruning 방법**

- 향상된 부분: 시공간 복잡도가 개선된다. tree의 branch 개수가 감소하기 때문에 각 branch마다 파생되는 추가적인 branch를 생략할 수 있으므로 계산하고 저장해야 하는 노드의 개수가 줄어들어서 프로그램의 시공간 복잡도를 개선시킬 수 있다.

- 향상되지 않은 부분: 가장 높은 점수를 획득할 수 있는 경로를 고려하지 못해 추천시스템을 통해 획득할 수 있는 점수가 감소할 수 있다.

2) **데이터의 단순화**

- 향상된 부분: 필드의 모든 칸에 대한 정보를 배열로 표현할 필요가 없으므로 프로그램에서 소모되는 공간 복잡도가 개선된다.

- 향상되지 않은 부분: 각 열에 존재하는 블록의 최대 높이만을 기억하므로 필드의 중간에 생기는 구멍에 대한 정보는 손실되어 추천시스템이 정확하게 작동하지 않을 수 있다.

전공: 국제한국학

학년: 4

학번: 20181202 이름 : 김수미

3. 테트리스 프로젝트 3주 과정을 통해 습득한 내용이나 느낀 점을 기술하십시오.

항상 입력-출력 형식의 프로그램만 작성하다가, 이렇게 실시간 반응형 게임 형식의 프로그램 코딩을 처음 해보았는데 재미있었다. 3주간 커리큘럼에 따라 프로젝트를 진행했는데, 그 과정 속에서 나중에 혼자 크기가 큰 프로그램을 기획하고 제작할 때 어떤 순서로 빌드업해 나가야 하는지에 대한 안목을 기를 수 있었다.