10주차 결과보고서

전공: 경영학과 학년: 3학년 학번: 20190963 이름: 한다현

**1.**

10주차 실습을 위해 사용한 자료구조는 RecNode 구조체이다. 이 구조체는 tree를 구성하는 노드이다. 구조체 내에는 노드의 레벨, 현재 노드까지 누적된 점수, field, child를 저장하는 배열, block ID, X좌표, Y좌표, 회전수, 부모 노드를 가리키는 포인터가 저장된다. tree의 root를 RecNode \*root로 선언하고 root를 동적 할당한다. root의 lv, accu\_score를 0으로, root의 f를 field로 초기화한다. 또한 누적 점수의 최대값을 의미하는 highest라는 변수를 0으로 초기화하고 recommend 함수를 실행한다.

recommend 함수는 두 개의 for loop을 실행한다. 첫 번째 for loop은 rotation의 수(4)만큼 반복한다. 두 번째 for loop은 x의 좌표를 통해 블록이 위치할 수 있는 모든 x좌표를 찾는 반복문이다. 이 반복문 내에서 먼저 CheckToMove 함수를 호출하여 X좌표가 WIDTH 범위 내에 있는지 검사하고, 그렇지 않은 경우 두 번째 for loop을 break한다. x좌표가 정상 범위 내에 있으면 새로운 노드 node를 동적 할당하고, node의 f도 동적 할당하여 부모 노드의 f를 복사한다. 그 외 node의 다른 모든 변수에 적절한 값을 저장한다. 모든 변수에 값을 저장한 후에는 while문을 실행한다. 이 반복문에서는 CheckToMove의 결과가 1인 경우 반복하고, node의 Y좌표를 1 증가시킨다. while 반복문이 종료된 후에는 AddBlockToField와 DeleteLine 함수를 호출하여 node의 accu\_score에 더한다. 누적 점수를 구한 이후에는 node의 level이 고려하는 블록의 개수보다 작은지 검사하여 작으면 recommend 함수를 호출하고, 작지 않으면 다시 node의 누적 점수와 hightest를 비교한다. 이 때 node의 누적 점수가 더 크면 hightest에 노드의 누적 점수를 저장하고 temp가 node를 가리키도록 한다. temp의 parent가 root가 아니면 temp는 temp의 parent로 이동한다. 이 반복문을 통해 첫 번째 블록의 node를 구하면 recommendR에 rotation 횟수를, recommendX와 recommendY에 각각 node의 X, Y좌표를 저장한다.

recommend 함수에서는 여러 개의 반복문이 존재한다. 첫 번째 반복문인 rotation을 결정하는 반복문은 0부터 3까지 반복한다. 두 번째 반복문은 x 좌표를 구하는 반복문인데, 이 반복문은 정해진 횟수만큼 반복하지 않는다. 세 번째 반복문은 y 좌표를 구하는 while문인데, 이 반복문도 정해진 횟수만큼 반복하지 않고 field의 상황에 의존한다. 마지막 반복문은 root의 child 노드를 찾는 while문이다. 이 반복문 역시 정해진 횟수만큼 반복하지 않는다. 따라서 recommend 함수의 시간 복잡도는 O(n^3)이다.

recommend 함수는 노드의 필드를 저장하기 위해 2차원 배열을 사용한다. 또한 recommend 함수 내에서 recommend 함수를 호출하는 재귀 함수이다. 따라서 recommend 함수의 공간 복잡도는 O(n^3)이다.

**2.**

이번 주차 과제에서 delete line을 우선적으로 고려하는 pruning tree 방법을 사용하였다. 이 방법은 실습 시간에 작성한 recommend 함수보다 더 적은 경우의 수를 계산하므로 프로그램의 실행 시간을 줄일 수 있다는 면에서 recommend 함수보다 향상되었다고 할 수 있다. 하지만 delete line이 발생하지 않는 경우에는 modified recommmend 함수가 기존의 recommend 함수와 동일한 계산을 수행하기 때문에 delete line이 발생하는 경우에만 더 향상된 프로그램 실행 시간을 확보할 수 있다.

**3.**

테트리스 3주차 과정을 수행하면서 linked list 사용에 더욱 익숙해졌다. recommend 함수를 작성하는 과정이 이전 주차 과정보다 더 어려웠는데, 이를 해결하고 나니 이전보다 linked list와 구조체에 대해 더 잘 이해할 수 있었다. 또한 알고리즘에 대해 많이 고민할 수 있었던 시간이었다. 특히 과제를 하면서 기존에 작성했던 recommend 함수를 어떻게 하면 더 효율적으로 수정할 수 있을까 고민하였고, 그 과정에서 여러 가지 알고리즘을 생각해보고, 직접 작성하며 경험할 수 있었다. 프로그램을 수행하는 시간과 프로그램 수행에 필요한 메모리 공간에 대해서도 배울 수 있었다. 이전에는 시간 복잡도나 공간 복잡도에 크게 신경을 쓰지 않았는데, 이번 주차 실습과 과제에서는 프로그램의 수행시간이나 메모리 공간에 대해 신경을 쓰면서 코드를 작성하였고, 프로그램의 수행 시간과 메모리 공간을 구하는 방법도 배울 수 있었다.

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명