CSE3013 (컴퓨터공학 설계 및 실험 I)
PRJ-1 테트리스 프로젝트 2주차 결과 보고서

서강대학교 컴퓨터공학과 박수현 (20181634)

서강대학교 컴퓨터공학과

1 목적

테트리스 게임에 랭킹을 추가한다.

2 문제

알고리즘의 의사 코드는 예비 보고서에서 작성한 코드와 동일하다. 실제 소스 코드에서는 메서드 이름에 BST[Binary Search Tree]나 RB-tree 등의 이름을 사용하지 않고 ordered list라는 표현을 사용했는데, 이는 자료 구조의 동작을 추상화하고 main.c에서 조금 더 이해하기 쉬운 코드를 작성할 수 있게 하려는 일환이다.

다음은 실습에서 구현한 함수들의 기능과 복잡도 일람이다.

함수명	기능	시간 복잡도	공간 복잡도
BST-LEFT-ROTATE	노드 x의 위치로 x의 오른쪽 자식을 옮긴다.	$\mathcal{O}(1)$	$\mathcal{O}(1)$
BST-RIGHT-ROTATE	노드 x의 위치로 x의 왼쪽 자식을 옮긴다.	$\mathcal{O}(1)$	$\mathcal{O}(1)$
BST-TRANSPLANT	u의 서브트리를 v의 서브트리로 대체한다.	$\mathcal{O}(1)$	$\mathcal{O}(1)$
BST-GET	특정 순위의 노드를 가져온다.	$\mathcal{O}(\log n)$	$\mathcal{O}(n)$
BST-INSERT	트리에 새 노드를 삽입한다.	$\mathcal{O}(\log n)$	$\mathcal{O}(1)$
BST-INSERT-REVALIDATE	트리에 새 노드를 삽입한 후, RB-tree의 조건을 충족하도록 트리를 고친다.	$\mathcal{O}(\log n)$	$\mathcal{O}(1)$
BST-DELETE	트리에서 노드를 삭제한다.	$\mathcal{O}(\log n)$	$\mathcal{O}(1)$
BST-DELETE-REVALIDATE	트리에서 노드를 삭제한 후, RB-tree의 조건을 충족하도록 트리를 고친다.	$\mathcal{O}(\log n)$	$\mathcal{O}(1)$
BST-QUERY	특정 순위 구간에 있는 노드들을 쿼리한다.	$\mathcal{O}(n)$	$\mathcal{O}(n)$

예비 보고서에서 BST-QUERY에 대한 $\mathcal{O}((r-l)\log n)$ 최적화를 언급했는데, 실제 코드에서 게임이 끝난 후 최대 5개까지의 원소만을 쿼리했다. 랭킹 리스트가 커지고 이 작업이 자주 일어난다면, 5개의 원소만을 쿼리할 때 $r-l=5$이고 이 때 이 메서드의 시간 복잡도는 $\mathcal{O}(\log n)$이 되고, 이는 n이 충분히 클 경우 일반적으로 $\mathcal{O}(n)$보다 효율적이므로 최적화가 수행된 메서드를 따로 만드는 것도 고려할 수 있다.

다음은 각 연산에 대한 정렬 상태의 연결 리스트와 RB 트리로 구현된 BST의 복잡도 비교 일람이다.

기능	정렬 상태의 연결 리스트		BST	
	시간 복잡도	공간 복잡도	시간 복잡도	공간 복잡도
원소 삽입	$\mathcal{O}(n)$	$\mathcal{O}(1)$	$\mathcal{O}(\log n)$	$\mathcal{O}(1)$
원소 삭제	$\mathcal{O}(n)$	$\mathcal{O}(1)$	$\mathcal{O}(\log n)$	$\mathcal{O}(1)$
쿼리	$\mathcal{O}(n)$	$\mathcal{O}(1)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$

따라서 쿼리할 때의 공간 복잡도를 제외하고 모든 면에서 일반적으로 BST가 정렬 상태의 연결 리스트보다 효율적이다.

3 습득한 내용

C++ STL의 `std::set`을 사용해 본 적은 많았으나 내부 원리를 자세하게 이해하고 구현할 수 있는 정도는 아니었는데, `std::set`의 내부적인 원리와 RB 트리를 *Introduction to Algorithms*을 참고해 이해하고 직접 구현해 볼 수 있는 기회가 되었다.

참고문헌

[1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*. Cambridge, MA: The MIT Press, 2009, pp. 287-338.