9주차 예비보고서

전공: 컴퓨터공학과 학년: 4학년 학번: 20202106 이름: OSHIMA ASUKA

1. 이번 주 실습에서 작성할 코드를 구현하기 위해 고려할 수 있는 배열과 LinkedList의 두가지 자료구조에 대해 생각해보도록 한다. 먼저 배열은 메모리상에서 연속하여 정보를 저장하기 때문에 인덱스를 이용하여 접근할 때 효율적이고, 특정한 요소에 쉽게 접근할 수 있다는 장점이 있다. 하지만 배열 사이즈는 고정되어 있어 변경하기 어렵다. 또 새로운 요소를 특정한 위치에 삽입/삭제하고자 하는 경우 요소들을 한 칸씩 shift할 필요가 있다는 단점도 있다.

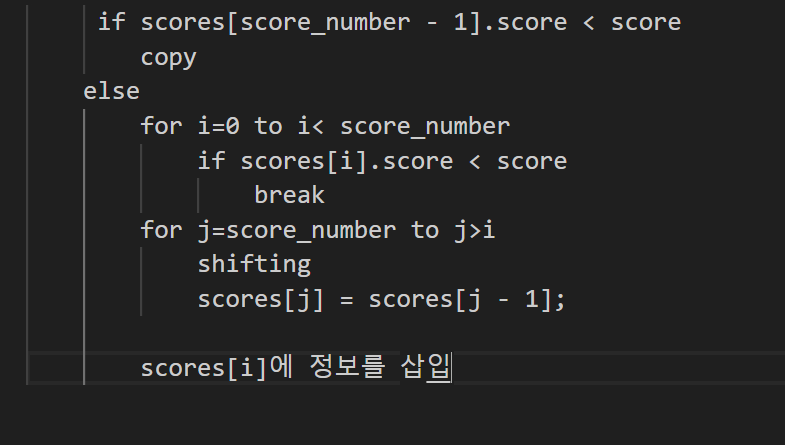
LinkedList는 동적으로 사이즈를 변경할 수 있고 요소의 삽입/삭제가 효율적이지만 특정한 요소에게 접근하려고 할 때 노드를 하나씩 방문해야 한다.

2. 랭킹을 삽입 구현하기 위한 pseudo code

* **Array**

시간 복잡도 : 삽입하려는 요소를 가장 앞으로 삽입할 경우 요소들을 다 한 칸씩 시프트해야 한다. 따라서 시간 복잡도는O(n)

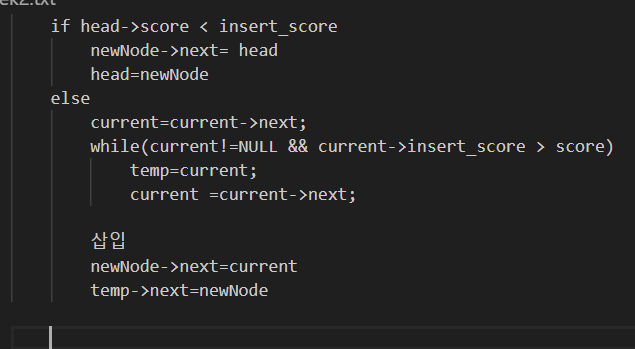
공간 복잡도:O(1).



* **Linked List**

시간 복잡도:　새로운 노드가 리스트의 맨 앞에 삽입할 경우 O(1). Worst case에서는 리스트 전체를 스캔할 필요가 있고 시간 복잡도는 O(n)이다.

공간 복잡도: O(1) 새로운 노드를 추가하기 위한 메모리양을 일정하다.

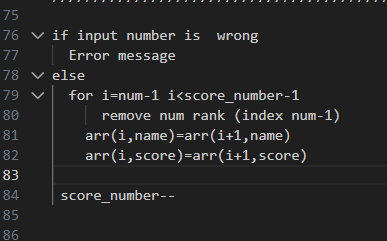


1. 랭킹을 삭제를 구현하기 위한 pseudo code

* **Array**

시간 복잡도: worst case O(n)이다. 배열의 가장 앞의 요소를 삭제하고자 하면 나머지 요소를 다 시프트해야 한다.

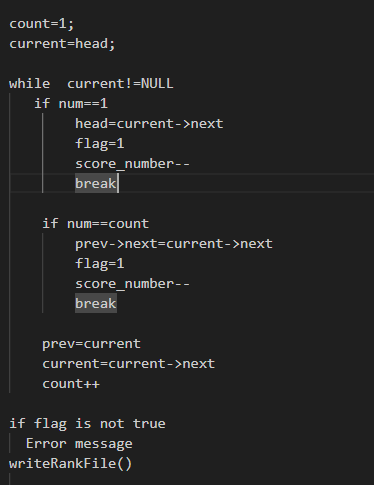
공간 복잡도: O(n).



* **Linked List**

시간 복잡도 : 삭제하고자 하는 요소를 찾기 위해 linkedlist 전체를 회해야 하기 때문에 O(n).

공간 복잡도: O(n).



1. Linked List의 노드를 루프를 돌리면서 하나씩 방문하면서 그 값이 출력해야 할 범위라면 출력하고 간다. 범위를 넘은 경우 루프부터 나간다. 시간 복잡도는 O(Y). 공간 복잡도는O(1).

