9주차 예비보고서

전공: 아트&테크놀로지학과 학년: 4학년 학번: 20191048 이름: 김도솔

**1.**

**Linked List**

랭킹 삽입 {

새로운 노드 생성

노드에 이름 저장

노드에 점수 저장

노드의 link를 NULL로 설정

현재 리스트에 노드가 없으면

first를 새로운 노드로 설정

현재 리스트에 노드가 있으면

리스트의 처음부터 끝까지 돌면서

점수 비교

새로운 노드의 점수가 가장 클 경우

새로운 노드의 link를 first로 설정

first를 새로운 노드로 설정

루프 탈출

그렇지 않으면

그 위치 바로 앞의 노드의 주소를 pre 변수에 저장

루프 탈출

새로운 노드의 link는 pre->link

pre->link는 새로운 노드

헤드 노드의 노드 수 +1

}

시간복잡도: O(n), 공간복잡도: O(1)

랭킹 삭제(num) {

node\* pre = first;

1부터 num - 2까지

pre = pre->link

pre가 NULL이면 오류

node\* del = pre->link

pre->link = pre->link->link;

free(del);

}

시간복잡도: O(n), 공간복잡도: O(1)

사용자가 부분적으로 확인하길 원하는 정렬된 랭킹(x~y위, x ≤ y, x, y는 정수)의 정보 얻기 {

int index = 0;

node\* curr = first;

curr이 null까지 curr = curr->link

index + 1이 x보다 크거나 같고 y보다 작거나 같으면

curr의 이름과 점수 정보 출력

index++;

}

시간복잡도: O(n), 공간복잡도: O(1)

**2.**

**Structure Array**

랭킹 삽입 {

랭킹에 삽입할 이름과 점수 입력 받기

i는 0부터 array length – 1 까지

점수가 arr[i].score보다 크거나 같으면 continue

그렇지 않으면 break

j는 i부터 array length – 1 까지

arr[j + 1].name = arr[j].name

arr[j + 1].score = arr[j].score

arr[i].name = 새로운 이름

arr[i].score = 새로운 점수

array length ++

}

시간복잡도: O(n), 공간복잡도: O(1)

랭킹 삭제(num) {

i는 num부터 array length – 2 까지

arr[i].name = arr[i + 1].name

arr[i].score = arr[i + 1].score

array length --

}

시간복잡도: O(n), 공간복잡도: O(1)

사용자가 부분적으로 확인하길 원하는 정렬된 랭킹(x~y위, x ≤ y, x, y는 정수)의 정보 얻기 {

i는 x에서 y까지

arr[i].name 출력

arr[i].score 출력

}

시간복잡도: O(n), 공간복잡도: O(1)