9주차 예비보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20221532 이름: 김민기

1.

- List 자료형

(rank 개수)\*2 의 2차원 배열을 사용한다. Txt 파일에는 내림차순으로 자료가 정렬되어 있으므로 순서대로 배열에 이름과 점수를 저장한다. List 자료형으로 자료구조를 구성하면 X, Y값을 받아 특정 구간의 Rank를 반환할 때 빠르게 접근이 가능하다. 하지만 중간에 위치한 점수를 삭제하거나 점수를 삽입할 때에는 모든 값을 옮겨야 하는 단점이 있다.

- Linked List 자료형

Node 구조체를 선언하여 내부에 이름과 점수, 다음 노드의 주소를 저장한다. 특정 위치의 rank를 접근할 때에는 첫 노드부터 탐색하므로 위의 List에 비해 시간 복잡도가 크지만 값을 삭제하고 삽입할 때에는 모든 노드를 옮길 필요가 없으므로 List에 비해 더 작은 시간 복잡도를 가진다.

2.

값을 옮길 필요가 없는 Linked List 자료형이 List자료형보다 평균적으로 적은 시간 복잡도를 지닌다.

- List 자료형

(1) 삽입

For(특정 위치부터 배열의 끝까지 반복)

특정 위치에 값을 삽입하고 한 칸 씩 뒤로 옮김

최악의 경우를 가정한다면 첫번째 위치에 값을 삽입하는 것이므로 시간 복잡도는 O(n)이다.

공간 복잡도는 스택을 사용하지 않으므로 O(1)이다.

(2) 삭제

For(특정 위치부터 배열의 끝까지 반복)

한 칸 뒤의 값을 앞의 칸으로 덮어씌움.

최악의 경우 첫번째 위치의 값을 삭제하는 것이므로 시간 복잡도는 O(n)이다.

공간 복잡도는 O(1)이다.

-Linked List 자료형

(1) 삽입

While(current\_node가 NULL이 아님)

If( cnt 가 특정 위치와 같음)

새. 노드를 사이에 연결해줌

Current\_node를 다음 노드로 이동

cnt++

최악의 경우 모든 노드를 검색하므로 시간 복잡도는 O(n)이다.

공간 복잡도는 O(1)이다.

(2) 삭제

While(current\_node가 NULL이 아님)

If( cnt 가 특정 위치와 같음)

앞과 뒤의 노드를 이어줌

특정 노드를 삭제

Current\_node를 다음 노드로 이동

cnt++

최악의 경우 모든 노드를 검색하므로 시간 복잡도는 O(n)이다.

공간 복잡도는 O(1)이다.

3.

평균적으로 List자료형이 특정 노드까지 검색하며 접근하는 Linked List보다 작은 시간 복잡도를 갖는다.

- List 자료형

For( i= X ~ Y)

List[i] 출력

최악의 경우 모든 리스트를 검색하므로 O(n)이다. 평균적으로

공간 복잡도는 O(1)이다.

- Linked List 자료형

While(current\_node가 NULL이 아님)

If( cnt 가 X와 Y 사이에 있음)

점수 출력

Current\_node를 다음 노드로 이동

cnt++

최악의 경우 모든 리스트를 검색해야 하므로 O(n)이다.

공간 복잡도는 O(1)이다.