**학과: 컴퓨터공학과 학년: 2 학번: 20191657 이름:최세은**

1. **2주차 실습에 구현하는 랭킹 시스템에 대한 자료를 읽어보고, 이를 구현하기 위한 다양한 자료구조를 2가지 이상 생각한다.**
2. 배열(array)

사용자의 이름과 점수를 저장할 때 이용한다. 새로운 정보를 저장할 때 해당 정보의 점수에 맞는 위치에 정보를 저장한다. 따라서 원래 있던 정보 중 새로운 정보보다 점수가 낮을 경우에는 한 칸씩 밀려나게 된다. 만약에 새롭게 저장할 때 배열이 꽉 찼으면 크기가 두 배인 새로운 배열을 선언해 기존의 배열 내에 있던 정보들을 옮겨준다. 반대로 랭킹을 삭제할 때는 해당하는 인덱스를 찾아 정보를 삭제한다. 또한 정렬된 랭킹을 얻을 때는 입력받은 x와 y에서 1씩 빼준 후 반복문을 통해 해당하는 값을 출력한다.

1. LinkedList

배열과 마찬가지로 사용자의 이름과 점수를 저장할 때 이용한다. 사용자가 게임을 종료했을 때 이름과 점수가 담긴 노드를 생성한다. 생성된 노드는 link를 따라 이동하면서 적절한 위치(점수가 내림차순으로 정렬되도록 하는 위치)를 찾으면 그 곳에 해당 노드를 삽입한다. 반대로 랭킹을 삭제할 때는 해당하는 노드를 link를 통해 이동하며 찾은 후 삭제한다.

1. **생각한 각 자료구조에 대해서 새로운 랭킹을 삽입 및 삭제하기 위해 필요한 시간 및 공간 복잡도를 계산한다.**
2. 배열(array)

새로운 랭킹을 삽입할 때는 삽입하는 곳 이후의 랭킹들은 한 칸씩 밀리게 되므로 현재 저장된 값들의 개수가 n이면 시간 복잡도는 O(n)이다. 그리고 값이 배열에 얼마나 있는지 상관없이 항상 배열의 크기는 일정하므로 배열의 크기를 m이라고 하면 공간 복잡도는 O(m)이다. 또한 랭킹을 삭제할 때는 입력받은 랭킹의 값에서 1을 빼주면 바로 index를 파악할 수 있으므로 시간 복잡도는 O(1)이고, 공간 복잡도는 삽입과 마찬가지로 O(n)이다.

1. LinkedList

새로운 랭킹을 삽입할 때는 최악의 경우 현재 존재하는 모든 노드들을 탐색해야 하므로 전체 노드의 개수를 n이라고 하면 시간 복잡도는 O(n)이다. 또한 공간 복잡도도 전체 노드의 개수를 반영하므로 O(n)이다. 그리고 랭킹을 삭제할 때도 최악의 경우 모든 노드를 탐색해야 하므로 시간 복잡도는 O(n), 공간 복잡도도 마찬가지로 O(n)이다.

1. **생각한 각 자료구조에 대해서 어떻게 정렬된 랭킹(x~y위, x≤y, x,y는 정수)을 얻을 수 있을지에 대해서 생각해보고, 그 방법에 대해서 기술하시오.**
2. 배열(array)

정렬된 랭킹을 얻기 위해서는 랭킹을 삭제할 때와 같이 입력받은 x와 y의 값에서 1을 각각 빼주면 해당 랭킹의 index가 되므로, 반복문을 통해 index 내에서의 랭킹들을 출력한다.

1. LinkedList

정렬된 랭킹을 얻기 위해서 새로운 변수를 선언한 후 0으로 초기화하고, 노드를 찾을 때 마다 1씩 증가시키며 x와 같아지면 그 때부터 y와 같아질 때까지 해당 정보를 출력한다.