# 9주차 예비보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20231632 이름: Jumagul Alua

1. 랭킹 시스템을 구현하기 위한 두가지 자료구조

- Array

요소를 연속적인 메모리 위치에 저장하므로 저장된 데이터에 효율적으로 액세스하고 조작할 수 있다. 데이터 로컬리티는 캐시 성능을 향상시켜 연속적인 요소들을 비연속적인 구조들보다 더 빠르게 읽게 한다. 하지만 배열이 만들어지면 그 크기는 고정되고 변경할 수 없습니다. 그러므로, 배열이 가득차거나 처음 크기보다 더 많은 요소를 추가했을때 더 큰 새로운 배열을 만들고 기존 요소를 배열에 복사해야 한다. 이거는 다른 방법들보다 효율성이 떨어진다.

- Linked List

Linked list은 데이터 요소(노드)가 포인터를 사용해 순차적으로 연결되어 있다. 연결 리스트는 array와 달리 노드를 추가하거나 제거함으로써 크기가 동적으로 변할 수 있다. 테트리스 게임에서 랭킹 시스템을 도입할땐 데이터를 산입하고 삭제하는 작업을 할 것이다. Linked List은 요소를 연속적으로 저장할 필요가 없기 때문에 요소를 효율적으로 삽입하고 삭제할 수 있다.

2. pseudo code 및 시간복잡도

Array을 이용해서 산입:

Function InsertAtIndex(index, len, score):

index = 0

while idx < len(arr) and arr[idx].score >= score:

```
        index++
for i = len(arr) - 1 to index + 1 step -1:
        arr[i].name = arr[i-1].name
        arr[i].score = arr[i-1].score
arr[index].name = name
arr[index].score = score
len++
```

Array을 이용해서 삭제:

```
function deleteAtIndex(arr, idx, len):
if index < 0 or index >= len:
        return
for i = index to len - 2:
        arr[i].name = arr[i+1].name
        arr[i].score = arr[i+1].score
len--
```

시간 및 공간 복잡도는 O(n)이다.

Linked List을 이용해서 산입:

```
function insertScoreAtPosition(score, position):
if position == 0:
        insertScoreAtBeginning(score)
else:
        newNode = createScoreNode(score)
        current = head
```

```
    for i=0 to position - 2:
        current = current->next
    newNode->next = current->next
    current->next = newNode
```

Linked List을 이용해서 삭제:

```
function deleteScoreAtPosition(position):
if position == 0:
    deleteScoreAtBeginning()
else:
    current = head
    for i from 0 to position - 2:
        current = current->next
    temp = current->next
    current->next = current->next->next
    free(temp)
```

시간 및 공간 복잡도는 O(n)이다.

3. 랭킹 정보를 얻는 방법

Array:

```
input x from user
input y from user
rankings = [ ]
for i=x to y:
    if i < len(sortedRankings):
```

```
        rankings.append(sortedRankings[i])
return ranking
```

시간 복잡도와 공간 복잡도는 O(y-x)이다.

Linked List:

```
input x from user
input y from user
for x=0 to y:
if(i>=x):
     current->score
current=current->next
```

시간 복잡도와 공간 복잡도는 O(y)이다.