

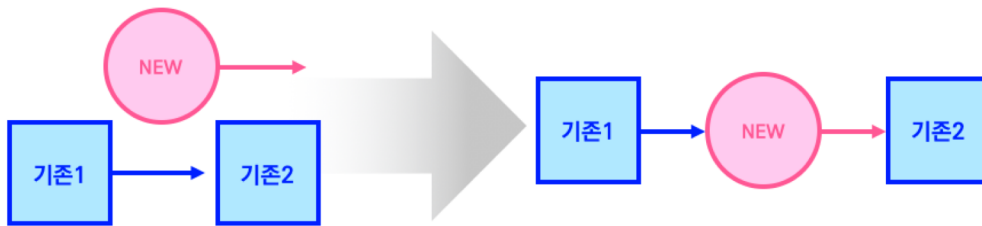
1. 실험 시간에 작성한 랭킹 시스템의 자료구조와 랭킹 시스템의 각 기능에 대한 알고리즘을 요약하여 기술하시오. 본인이 선택한 랭킹 시스템을 구현하기 위한 자료구조가 왜 효율적인지 시간 및 공간 복잡도를 통해 보이고 설명하시오.

1) 랭킹 시스템의 자료구조

데이터 필드(사용자 이름과 점수를 저장)와 포인터 링크 필드(다음 노드를 가리키는 역할)로 구성된 노드 구조체 변수들을 연결하여 랭킹 데이터를 관리하는 Linked List(연결 리스트) 자료구조를 이용했다.

2) 랭킹 시스템의 각 기능에 대한 알고리즘

- 랭킹 정보 등록 기능 : 연결 리스트는 점수가 큰 노드부터 작은 노드 순으로 차례대로 이어져 있다. 따라서 가장 큰 점수 값을 저장하고 있는 헤더 노드부터 차례대로 포인터를 따라가면 저장된 점수를 큰 것부터 작은 것 순서대로 모두 확인할 수 있다. 따라서 새로운 점수가 들어오면 노드를 순서대로 탐색하면서 해당 점수가 들어갈 수 있는 자리에 값을 넣어주기만 하면 된다.



- 랭킹 정보 확인 기능 : 랭킹을 저장하는 노드는 한 개씩 1등부터 꼴등까지 이어져 있기 때문에 1등부터 차례대로 훑으면서 원하는 등수의 랭킹 정보를 확인하면 된다. 노드에 등수 관련 데이터가 저장되어 있지 않더라도 앞에서부터 확인한 노드의 개수를 세기만 해주면 각 노드가 몇 등에 해당하는 노드인지를 알 수 있다.

3) 연결 리스트 자료구조가 효율적인 이유

Game over가 되고, 새로운 랭킹 정보(사용자 이름, 점수)가 등록될 때, 시간 및 공간 복잡도

>> 기존에 저장되어 있던 랭킹의 개수가 N개 라고 할 때, Worst case의 시간복잡도는 $O(N)$ 이고 공간복잡도는 $O(N+1)$ 로 간단하게 $O(N)$ 으로 나타낼 수 있다. N개의 노드보다 새롭게 삽입되는 랭킹의 값이 작다면 가장 마지막 자리에 새로운 노드가 삽입되어야 하므로, 총 N노드를 훑은 다음에야 삽입이 가능하고, 총 N+1개의 노드를 저장해야 하기 때문이다.

원하는 랭킹 정보를 입력 받고, 랭킹을 추출하기 위한 과정에서 자료구조를 탐색 및 랭킹 추출에서의 시간 및 공간 복잡도

>> 현재 저장되어 있는 랭킹의 개수가 N개라고 할 때, Worst case의 시간복잡도는 $O(N)$ 이고 공간복잡도 역시 $O(N)$ 이다. 가장 마지막 랭킹을 추출하기 위해선 모든 노드를 한번씩 훑어야 하고 총 N개의 노드를 저장해야 하기 때문이다.

프로그램의 시공간 복잡도가 exponential 하게 증가하지 않고 linearly 증가하기 때문에, log단위의 시공간 복잡도 보다는 아니겠지만 어느 정도 우수한 효율성을 가진다고 이야기할 수 있다.

2. 본 실험 및 숙제를 통해 습득한 내용을 기술하시오.

- 실험에서 연결 리스트를 직접 구현하여 사용자 랭킹 정보 관리에 이용함으로써 새롭게 생성되는 데이터를 효율적으로 관리하는 방법을 연습해볼 수 있었다. 숙제를 통해서는 데이터를 추가하는 것 뿐만 아니라 삭제하는 방법을 익힐 수 있었고, 단순히 순서가 아닌 이름으로 연결 리스트의 노드를 탐색하는 기능 또한 구현해 볼 수 있었다. 그 과정에서 여러가지 예외 상황을 관리하는 것이 중요하다는 것을 깨달았고, 이를 위해 사전에 플로우 차트와 수도코드를 꼼꼼하게 작성하는 것의 중요성을 느낄 수 있었다.