9주차 결과보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20211522 이름: 김정환

**1.**

.................

주어진 랭킹 시스템에 대해서는 Linked List를 이용하여 구현하였다. Linked List에서 각 노드는 구조체를 통해서 char type의 array인 name 변수와, int type의 score 변수, 그리고 다음 노드를 가리킬 Node\* type의 link로 구성되도록 하였다. Linked List는 탐색이나 삭제에 있어서 최대 O(n)의 시간 복잡도를 가지고, 또한 삭제의 구현이 간단히 할 수 있다. 또한 ranking에 따라서 순서대로 저장할 때 내림차순 또는 오름차순으로 저장하기에 적합하여 효율적이라고 생각했다. 특히 크기에 따라 노드의 연결만 바꾸면 되기에 크기에 맞춰 새 노드를 넣을 때도 최악의 경우 O(n)의 시간 복잡도를 가져 효율적이라 판단했다. 구현한 4개의 함수에 대해서는 각각의 기능이 있다. 먼저 createRankList() 함수는 rank.txt로부터 rank 데이터를 받아와서 구성한 Linked List에 제대로 저장될 수 있도록 하는 함수이다. 주어진 rank.txt 파일은 점수가 내림차순으로 이미 정리된 채로 주어지므로, score\_number에 rank의 개수를 입력받고 난 후에는 그대로 head가 비어있을 때만 구분해주고 입력받아 link로 계속 연결하면 된다. rank() 함수는 메뉴를 선택하는 창에서 2번 rank를 선택했을 때 나와야 할 기능들을 구현한 함수이다. 우선 while loop 속에서 ch에 ‘1’ ~ ‘3’ 중에서만 입력이 나올 때까지 입력을 받는다. 입력을 받았을 때 ‘1’의 경우에는 X, Y를 입력받아 X위부터 Y위까지 출력한다. 우선 입력이 없는 경우에 대해서 X = 1, Y = score\_number가 되도록 처리했다. X가 Y보다 큰 경우 역시 예외로 처리했다. 이 후 순위에 맞는 출력은 idx 변수를 통해 개수를 세는 방법으로 구현했다. X번까지 link로 이동하고, 그 순서부터 Y-X + 1번 for문으로 출력한다. 이 과정에서 최악의 경우에는 리스트의 처음부터 끝까지 탐색하는 것이므로 리스트의 길이를 n이라 하면, 시간 복잡도는 O(n)이고, 공간 복잡도 역시 O(n)이다. ch가 ‘2’인 경우에는 str에 name을 입력 받고, 이를 통해서 랭킹에서 찾아서 출력한다. List를 전체 탐색하여 같은 경우에 출력한다. 한 번도 같지 않아 check가 그대로 0인 경우만 오류를 출력하도록 하였다. 이 경우에는 무조건 List를 전체 탐색하므로 시간 복잡도는 O(n), 공간 복잡도는 O(n)이다. 텍스트, 스크린샷, 디스플레이, 폰트이(가) 표시된 사진

자동 생성된 설명

ch가 ‘3’인 경우에는 rank를 입력 받고, 우선 범위 밖인지 체크하여 list에 있는지 판단한다. 그 후 범위 내에 있는 경우에는 ‘1’일 때와 유사하게 맞는 rank를 찾는다. 차이점으로는 pre 변수에 ptr 이전의 노드를 저장하면서 이동하는 점에 차이가 있다. pre의 link에 삭제할 노드인 ptr의 link를 연결하고, ptr은 free로 메모리를 해제하면서 삭제한다. 또한 score\_number를 1 감소하고, writeRankFile()을 실행하여 rank.txt에도 반영되도록 조정한다.

ptr

head

노드1

삭제할 노드

다음 노드

pre

ptr

head

노드1

삭제할 노드

다음 노드

ptr

pre

head

노드1

삭제할 노드

다음 노드

head

노드1

다음 노드

삭제할 노드

다음 노드

head

노드1

삭제 알고리즘의 경우에도 최대 탐색은 제일 마지막에 있는 노드일 경우 이므로 최악의 경우에 대한 시간 복잡도는 O(n)이고, 공간 복잡도는 O(n)이다. 이에 따라 각각의 부분에 대해서 모두 최대 O(n)의 시간 복잡도이고, 최대 O(n)의 공간 복잡도이므로 rank()의 시간 복잡도와 공간 복잡도는 O(n)이다.

writteRankFile()은 rank.txt에 score\_number와 리스트에 있는 rank 정보를 저장하는 함수이다. while문을 통해서 ptr이 NULL이 나와서 리스트 끝까지 나올 때까지 fprintf로 file에 입력하였다. 이는 리스트의 처음부터 끝까지 탐색하면서 file에 기록하므로 O(n)의 시간 복잡도와 공간 복잡도를 가진다.

newRank() 함수는 gameover 시 사용자의 이름을 입력 받아 점수와 함께 리스트에 추가하여 랭킹에 추가하는 함수이다. 이름을 입력 받고, Linked List에 노드를 삽입하는 부분을 구현했다. head가 NULL일 때는 여기서 생성한 newnode를 head로 하고, head->link가 NULL인 경우에는 크기를 비교하여 newnode가 더 크면 head로 바꾸고 아닌 경우에는 head에 연결한다. 그 외의 경우에 대해서는 연결하는데 있어서 head의 score보다 큰 경우만 예외 처리를 해주었다. 해당 경우에는 newnode의 link를 head로 하고, head를 newnode로 바꿔주어 newnode가 head가 되도록 했다. 이 외에는 newnode가 더 클 때까지 while loop로 탐색하고, 저장한 pre의 link를 newnode, newnode의 link를 ptr로 해서 연결한다. 해당 함수는 최악의 경우가 역시 리스트의 끝까지 탐색하는 것이므로 시간 복잡도 O(n), 공간 복잡도 역시 O(n)이다.

................

**2.**

.......................

Ranking system에 대한 구현을 하면서 Linked List에 필요한 기능들에 대한 실질적 구현에 대해서 많이 연습할 수 있었다. insert에서 조건을 넣어 적절히 정렬된 리스트를 만드는 연습이 되었다. 또한 ncurses library에서 echo()와 noecho()로 입력이 보였다 안 보였다 하게 하는 기능에 대해서 알 수 있었다. 또한 파일에 쓰는 부분을 구현할 때 while에 조건을 넣었을 때와 무한으로 돌리고, NULL일 때 break가 되도록 하는 부분에서 파일에 입력이 되는 경우와 안 되는 경우에 대해서 새로 알 수 있어서 파일에 입력 시 주의해야 할 부분에 대해서도 알 수 있었다.

.........................