9주차 결과보고서

전공: 아트엔테크놀로지 학년: 3학년 학번: 20191172 이름: 함승우

1. 선택한 자료구조의 효율성에 대해서 평가하고 왜 그렇게 생각하는지 상세하게 기술한다. 이 때 시간 및 공간 복잡도를 통해 보이고 설명하시오.

우선, 예비보고서에서 자료구조를 2가지 소개했다. array와 linkedlist이다. 실험은 linked list를 사용하여 진행을 했다. 우선 linked list를 선택한 이유는 array에 비해서 insert와 delete가 자유롭기 때문이다. 또한 array는 한정된 크기의 입력값(유저의 수)를 가질 수밖에 없다.

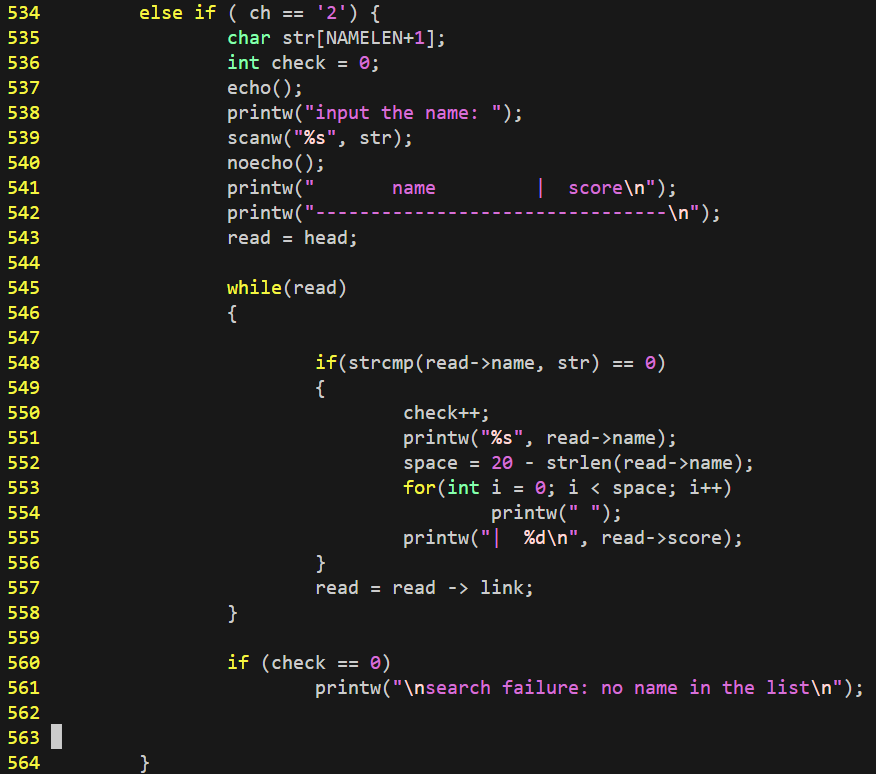
예를 들어보자면, 유저의 수는 제한이 되어있지 않는 테트리스 게임 프로그램에서 현재 유저가 n명까지 이 게임을 실행했다면, linked list의 공간 복잡도는 O(n)이라고 할 수 있다. 하지만, 유저의 수가 제한이 되어있지 않기 때문에 array의 경우 n개 값보다 크게 설정해야 한다. 이 경우 array의 공간 복잡도는 O(N)이 된다. N(N >> n)

각 자료구조에서 insert를 설명하자면, array에서의 insert는 array의 크기에 제한이 없는 한 중간에 값을 삽입하는 경우에는 linked list는 특정 node들의 link값을 수정하기만 하면 되는 것과는 달리 array는 중간에 삽입하려고 하는 위치 이후의 배열의 값들을 한 칸 미루고, 삽입을 진행해야 한다. 이는 매우 비효울적이다. Rank를 수정함에 있어 rank의 score 순서대로 자료구조를 수정할 때에는, array와 linkedlist는 서로 값을 비교하며 수정할 부분을 찾고 삽입해야 하므로 시간 복잡도는 O(자료구조의 크기)로 같을 수 있지만, 앞서 말했듯이 유저의 수가 제한이 없는 경우를 array로 구현하기 위해서는 너무 큰 size의 array가 필요해 data면에서 비효율적아며, array는 매번 한 칸 미루고 삽입하는 행위가 너무 비효울적이라고 볼 수 있다.

2. 랭킹 시스템의 각 기능에 대한 알고리즘을 요약하여 기술한다.

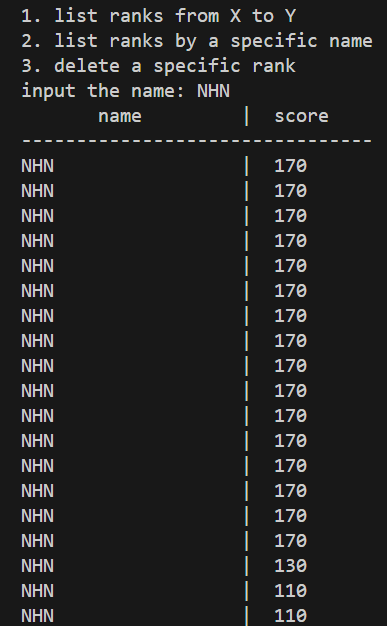
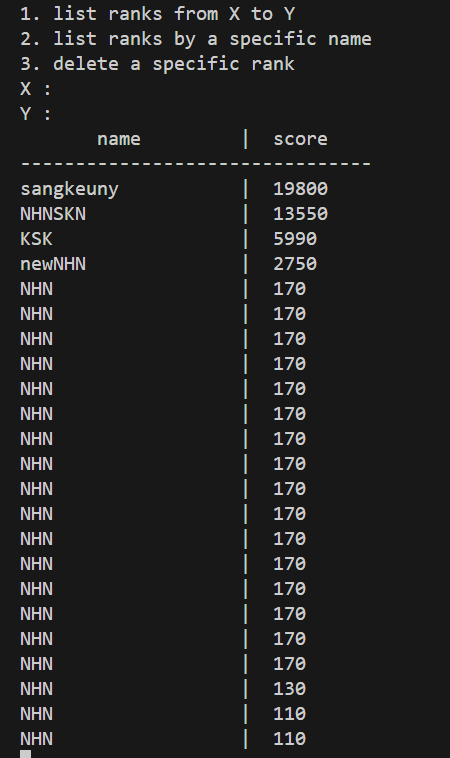
createRankList(0 함수에서는 rank.txt 파일을 읽어온 후, 이 rank.txt. 파일을 linked list에 저장하는 함수이다. createRankList() 함수에서는 insert() 함수를 호출하는데, insert 함수는 linked list의 맨 마지막 부분에 삽입한다. Rank() 함수는 입력한 값 1, 2, 3으로 정하는데, 각각 ranking을 어떻게 print할지에 대한 함수이다. newRank() 함수는 게임이 종료가 되었을 때 이 유저의 이름을 받아서 rank에 저장하고, 이 rank들을 순서대로 sorting 하는 함수이고, 이 rank들을 writeRankFile()을 통해서 rank.txt에 update를 한다.

3. rank() 모드 2: 사용자의 이름 입력



사용자의 이름을 검색하는 코드는 이렇다. 출력화면을 보이자면,

우선 원본 화면은



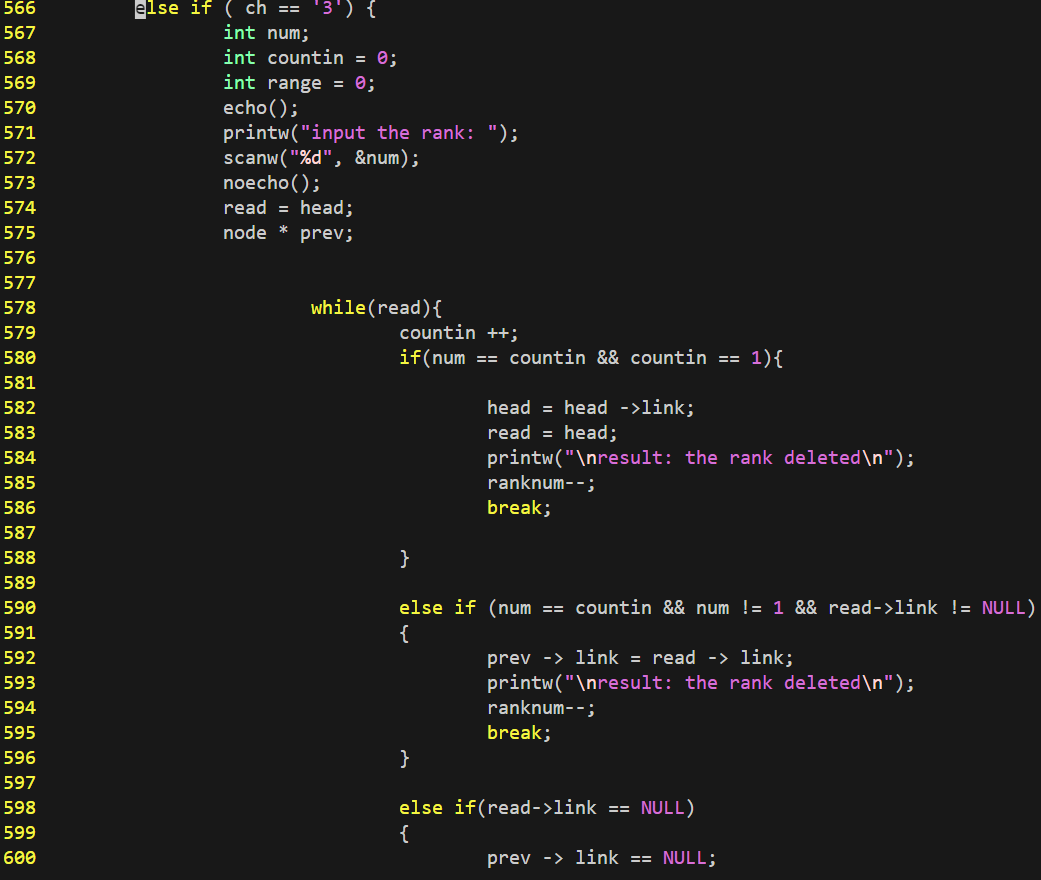
텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

왼쪽 상위 그림은 1번 모드에서 아무것도 입력하지 않아 원본 rank.txt.를 볼 수 있는 화면이고

오른쪽 상위 그림은 2번 모드에서 NHN이라는 이름을 검색해 나오는 이름 및 스코어를 출력하는 화면이다. 마지막 그림은 rank.txt에 없는 파일일 때 “search failure: no name in the list”라는 문구를 출력한다.

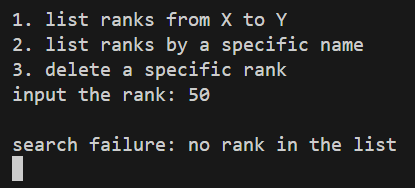
4. rank() 모드 3: 원하는 랭킹 정보 삭제



텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

왼쪽 상위 이미지부터 그림1, 그림2

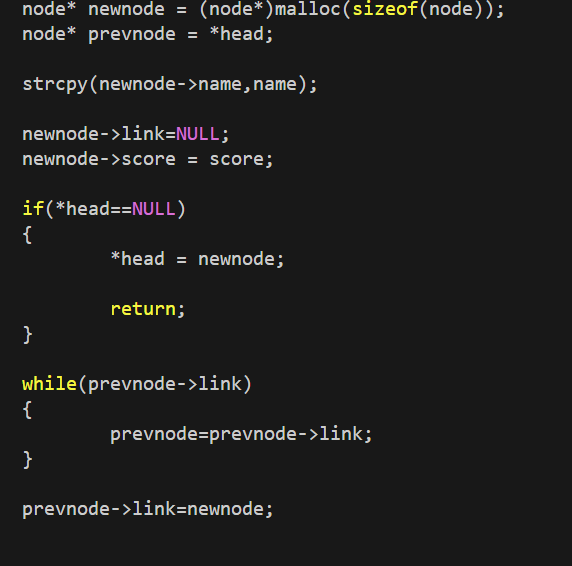
왼쪽 하위 이미지부터 그림3, 그림4

그림1과 그림2를 실행하면, rank가 1인 부분만 삭제되고 ranknum을 초과하는 숫자가 입력되었을 때는 삭제되지 않게 된다.

그림 3과 그림 4에서 1번째 요소만 삭제된 것을 확인할 수 있다.

5. 본 실험을 통해 습득한 내용을 기술하시오.

실습 기간 중 제일 힘들었던 부분은, createRankList()인데.



node를 삽입하는 과정에서 prevnode라는 node를 설정하고 이를 linked list의 맨 마지막(tail)로 보내서 그 prevnode에 newnode를 연결하려는 code를 만들려고 했는데, 저 while문의 순서를 다르게 해서 prevnode가 설정되지 않은 채 코드가 진행되어서 segmentation fault가 일어나게 되었다.

두 번째는, array와 linked list의 효용성에 대해서 습득했다. 알고리즘 문제를 풀 경우 data의 개수가 주어져 array를 쓰는 경우가 많았는데, linked list를 사용하면 매번 insert를 해야 하는 프로그램에서 이점을 얻을 수 있다는 사실을 알게 되었다.