

CSED232 Object-Oriented Programming

Spring 2019

Programming Assignment #1

Problem Solving

무은재 새내기학부

20180038

박형규

POVIS ID : hyeongkyu

Honor Code : 나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.

1. 프로그램 개요

이번 과제에서는 객체 지향 프로그래밍을 배우기 위해 필요한 기본적인 C언어 지식들의 숙련도와 문제를 해결하기 위한 구현능력을 점검해 보기로 한다. 1번 문제는 일렬의 나무막대기가 있을 때 왼쪽에서 보았을 때와 오른쪽에서 보았을 때 보이는 나무 막대기의 개수의 합을 구해주는 프로그램이다. 2번 문제는 일렬의 나무막대기가 있을 때 나무 막대기가 가장 많이 보일 수 있도록 나무 막대기를 제거하여 최대로 보이는 개수를 구해주는 프로그램이다. 3번 문제는 사용자가 입력한 거대한 빙하가 y 년 후에 몇 덩이의 빙하로 존재하는지 구해주는 프로그램이다.

2. 전체 구조

1번 문제는 왼쪽과 오른쪽에서 각각 배열의 원소들을 비교해 나가면서 가장 큰 원소 값을 설정하여 나무 막대기의 길이를 비교한다.

2번 문제는 LIS 알고리즘을 이용하여 새로운 배열 하나를 더 생성하여 새로운 배열의 마지막 원소보다 클 경우에는 새로운 배열 끝에 추가하고, 마지막 원소보다 크지 않을 경우에는 새로운 배열 안에서 크기순으로 들어갈 위치에 그 원소보다 더 큰 원소 대신에 넣는다.

3번 문제는 BFS 알고리즘을 이용하여 2차원 배열에서 각 원소를 검사하는 과정에서 각 원소와 인접해 있는 4개(혹은 2개 or 3개)의 다른 원소들도 함께 검사한다. 이 과정에서 queue를 사용하여 queue가 모두 비었을 때 섬의 개수가 하나씩 늘어나는 것으로 하였다. queue 사용을 위하여 pos, node, QUEUE 구조체를 사용하였는데, pos는 node안에 위치 정보를 담기 위한 간단한 구조체로 가로, 세로 좌표의 정보를 담는데 사용한다. node 구조체는 queue안을 구성하는 구조체로 위치 정보를 담고있는 ps, 섬의 높이를 담고 있는 data, 다음 노드의 위치를 담고 있는 next가 구성 요소로 있다. 마지막으로 QUEUE구조체는 전체 queue를 관리할 수 있는 구조체로 가장 앞의 node를 가르키는 front, 가장 마지막 node를 가르키는 rear, queue안의 노드의 개수를 나타내는 count가 구성요소이다.

3. 프로그램 실행 방법

1번 문제와 2번 문제는 모두 첫 번째 줄에 나무 막대기의 개수를 입력하고 두 번째 줄에 각 나무 막대기의 높이들을 입력하면 출력값이 나온다.

3번 문제는 첫 번째 줄에 격자의 크기 W(가로), H(세로), YEAR(시간)을 입력하고, 두 번째 줄부터 2차원 배열의 형태로 빙하의 높이들을 입력하면 출력값이 나온다.

4. 프로그램 실행 예제

```
7
4 6 5 8 6 7 1
6
C:\Users\박형규-PC\source\repos\Wassn1_1\Debug\Wassn1_1.exe(8676 프로세스)이(가) 0
코드로 인해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

1번 문제 실행 예제

```
7
4 6 5 8 6 7 1
4
C:\Users\박형규-PC\source\repos\Wassn1_2\Debug\Wassn1_2.exe(7832 프로세스)이(가) 0
코드로 인해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

2번 문제 실행 예제

```
5 5 4
1 5 4 8 1
2 3 7 9 2
1 5 1 2 6
4 2 7 8 4
9 8 3 6 5
6
C:\Users\박형규-PC\source\repos\Wassn1_3\Debug\Wassn1_3.exe(8256 프로세스)이(가) 0
코드로 인해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

3번 문제 실행 예제

5. 토론, 결론 및 개선방향

이번 과제를 진행하면서 약 2개월 동안 다루지 않았던 C언어를 다시 한 번 복습하며 프로그래밍을 하는 기초적인 능력을 다잡게 되었고, 특정 문제가 주어졌을 때 그 문제를 해결하기 위해 다양한 프로그래밍적 기술들을 활용하는 능력도 길렀다. 또한 이번 과제를 해결하기 위해 힌트로 주어진 LIS 알고리즘, BFS 알고리즘 등을

인터넷을 이용하여 찾아보다가 ‘시간 복잡도’라는 개념에 대해 알게 되었다. 지금까지 코딩을 하면서 코딩 후 결과물을 얻을 때까지의 걸리는 시간에 대해 생각해본 적은 단 한 번도 없었는데 프로그램의 코딩 방법에 따라 코딩 후 결과가 나올 때까지의 시간이 달라진다는 점에서 신기하였다. 또한 어떤 방법을 써야 시간 복잡도가 낮아지는지에 대한 고찰도 함께 해보게 되었다. 이번 과제에서는 시간복잡도가 $N\log N$ 인 방법을 사용하였는데, 이보다 시간 복잡도가 더 줄어들 수 있는 코딩 방법이 있는지 궁금하여 후에 추가적으로 고찰하여 볼 예정이다. 주어진 문제 이외에 추가적으로 적용할 수 있을 것 같은 기능은 1번 문제와 2번 문제에서는 각각 보이는 막대기의 번호, 최대로 보이게 하는 막대기의 번호를 알아낼 수 있을 것 같다. 또한 3번 문제에서는 어떤 섬이 가장 넓은 넓이를 가지고 있는지, 어떤 섬이 가장 높은 높이를 가지고 있는지를 알아낼 수 있을 것 같다.

6. 참고문헌

<https://jason9319.tistory.com/113>

<https://mygumi.tistory.com/72>

<https://mygumi.tistory.com/69>

<https://mygumi.tistory.com/303>

https://github.com/hotehrud/acmicpc/blob/master/algorithm/graph/1260_AdjacencyList.java

<https://www.acmicpc.net/problem/4963>

<https://romanceboong.tistory.com/entry/%EB%B0%B1%EC%A4%80-4963%EB%B2%88-%EB%AC%B8%EC%A0%9C-%EC%84%AC%EC%9D%98-%EA%B0%9C%EC%88%98>

<https://hyeonstorage.tistory.com/349>

<https://sarah950716.tistory.com/13>

<https://kwanghyuk.tistory.com/m/74>