

CSED232 Object-Oriented Programming (Spring 2019)

Programming Assignment #1

Problem Solving

Due data: 2019년 3월 8일 금요일 23:59

담당조교: 정진웅 (jinwoong.jung@postech.ac.kr)

본 과제에서는 학생들의 Problem Solving / Code Implementation 능력을 확인하기 위해 3개의 문제를 출제하였다.

채점 방식

Easy 난이도 / Hard 난이도의 2가지 dataset이 존재하며 점수 배점은 60% / 40%로 한다.

각 난이도 별로 정답을 맞춘 test case에 대해 부분 점수를 인정한다.

모든 test case에 대하여 10초 내에 결과가 출력되어야 한다. (초과시 0점)

모든 input과 output은 STDIN과 STDOUT을 사용한다. (파일 입출력 X)

각 문항별로 적힌 배점에 보고서 점수 10점을 더해 총 100점 만점으로 한다.

제출 방식

제출 코드는 문항별로 prob_1, prob_2, prob_3 폴더를 만든다.

제출 보고서는 학번.pdf로 만든다.

코드와 보고서 파일 전체를 학번.zip 파일로 묶어서 제출한다.

제출 방식을 지키지 않을 경우 총점에서 10% 감점

주의사항

동적 메모리 할당 사용시, 할당 해제를 제대로 하지 않을 경우 해당 문제 10%감점

Compile이 안되거나 실행이 안될 경우 0점 (minGW상에서 g++ compiler 사용)

보고서와 각 코드 main 위에 (주석으로) Honor code를 명시해야 함 (없을 시 0점)

Cheating은 0점, 참고한 자료 및 사이트는 보고서에 reference로 적는다.

문제에 따라 100만 크기의 array를 사용해야 하는데, local variable로는 해당 크기의 array가 잡히지 않는다. 본 과제에 대해서만 global variable의 사용을 권장함

일반적인 프로그래밍의 경우에는 global variable 사용이 권장되지는 않으나, 본 과제에 대한 프로그래밍의 경우 global variable 없이 문제를 푸는 것이 어려울 수 있음

C++에서 C의 Input, Output 사용

cstdio 파일을 include한다.

printf -> std::printf, scanf -> std::scanf로 변환하여 사용한다

예시

```
#include <cstdio>

int main()
{
    int var;

    std::scanf("%d",&var);
    std::printf("Hello World!\n");
    std::printf("%d\n",var);

    return 0;
}
```

위 프로그램은 int형 변수를 1개 입력받고, "Hello World!"문구와 입력 받은 변수를 출력한다.

```
123
Hello World!
123
```

기타

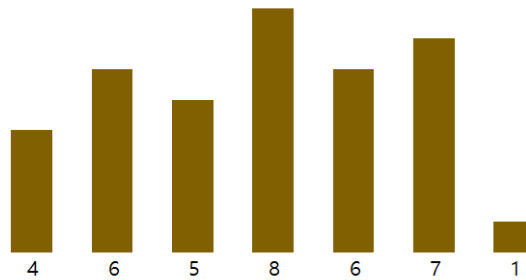
C++를 알고 있는 학생들은 프로그램을 C++로 작성해도 좋습니다. C++를 아직 모르는 학생들은 C를 사용하면 됩니다.

이미 STL을 알고 있는 경우에는 사용 가능하고 (e.g., std::queue, std::vector), 모르는 경우에는 그냥 array를 사용하면 됩니다.

잘못된 경우의 입력은 주어지지 않습니다. 예외처리를 할 필요는 없습니다.

1. 보이는 나무막대기 개수 세기 (배점: 40점)

아래 그림과 같이 서로 다른 길이를 가진 나무막대기가 1열로 늘어서 있다. 이때 왼쪽에서 보았을 때 보이는 나무막대기의 개수와 오른쪽에서 보았을 때 보이는 나무막대기의 개수의 합을 구하시오. (두 나무막대기의 높이가 같은 경우 가려져서 보이지 않는 것으로 함)



왼쪽에서 볼 경우 4, 6, 8의 세 막대기가 보이고, 오른쪽에서 볼 경우 1, 7, 8의 세 막대기가 보인다. 따라서 답은 6이다.

입력

첫번째 줄에 나무막대기의 개수 n 이 주어진다. (Easy: $1 \leq n \leq 100$, Hard: $1 \leq n \leq 1,000,000$)

두번째 줄에 각 나무막대기의 높이 h_i 들이 주어진다. ($1 \leq h_i \leq 1,000,000$)

출력

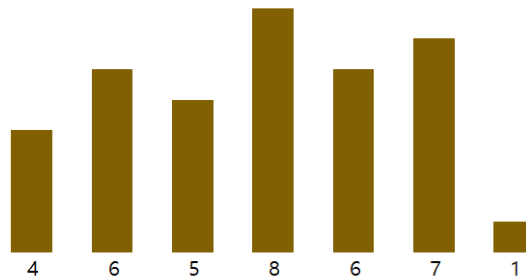
왼쪽, 오른쪽에서 보았을 때 보이는 나무막대기의 개수의 합을 출력한다.

예시

Input	Output
7 4 6 5 8 6 7 1	6

2. 보이는 나무막대기 개수의 최댓값 (배점: 20점)

아래 그림과 같이 서로 다른 길이를 가진 나무막대기가 1열로 늘어서 있다. 이때 왼쪽에서 보았을 때 보이는 나무막대기의 개수를 최대로 만들고 싶다. 막대기를 원하는 만큼 제거하여 최대로 보이는 개수를 구하시오. (나무막대기의 순서는 바꿀 수 없고, 제거만 가능함)



1, 3, 5, 6번째 나무막대기만 남길 경우 최대로 많이 보이게 된다. 따라서 최대로 보이는 나무막대기의 개수는 4개 이다.

입력

첫번째 줄에 나무막대기의 개수 n 이 주어진다. (Easy: $1 \leq n \leq 100$, Hard: $1 \leq n \leq 100,000$)

두번째 줄에 각 나무막대기의 높이들이 주어진다. ($1 \leq h_i \leq 1,000,000$)

출력

최대로 보이는 나무막대기의 개수를 출력한다.

예시

Input	Output
7 4 6 5 8 6 7 1	4

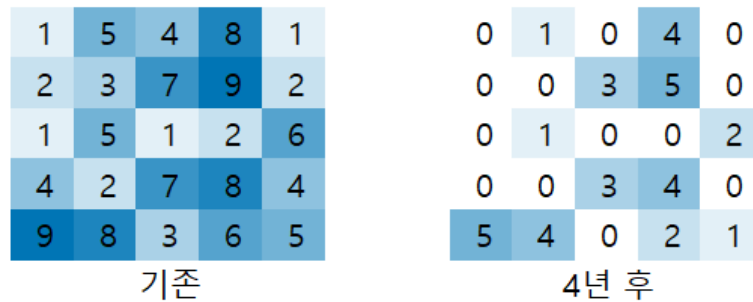
힌트

이 문제는 주어진 number sequence 에 대하여 longest increasing subsequence 를 찾는 문제와 밀접한 관계가 있다. 인터넷에서 longest increasing subsequence 에 관한 자료를 쉽게 찾을 수 있다. AssnReadMe 에 언급된 것처럼, Reference 에 정확히 언급하기만 하면 인터넷에 있는 자료를 숙제에 활용해도 된다.

3. 빙하의 개수 세기 (배점: 30점)

아래 그림과 같이 $W \times H$ 격자에 거대한 빙하가 존재한다. 빙하는 처음에는 한 덩이로 되어 있지만, 지구온난화에 의해 높이가 1년에 1씩 줄어들게 된다. y 년 후에 빙하가 몇 덩이 존재하는지 구하시오.

변이 맞닿아 있는 빙하만 같은 덩이로 친다. 대각선에 있는 빙하는 다른 덩이로 구분된다.



기존에는 모든 빙하가 같은 덩이에 있지만, 4년이 지난 후에는 6덩이로 쪼개지게 된다.

입력

첫번째 줄에 격자의 크기 W (가로), H (세로)와 시간 y (년)가 주어진다.

(Easy: $1 \leq W, H \leq 100$, Hard: $1 \leq W, H \leq 1,000$), ($0 \leq y \leq 100$)

두번째 줄부터 H 줄에 걸쳐 각 빙하의 높이 h_{ij} 들이 주어진다. ($1 \leq h_{ij} \leq 100$)

출력

y 년이 지난 후 존재하는 빙하의 개수를 출력한다.

예시

Input	Output
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

힌트

이 문제는 graph 에서 connected component 를 찾는 문제와 밀접한 관계가 있다. Graph 에서

connected component 를 찾기 위하여 DFS (Depth First Search) 또는 BFS (Breadth First Search)를 이용할 수 있다. 그런데, 주어진 문제는 구조가 간단하므로 굳이 graph 를 만들지 않고, 주어진 격자 구조에 DFS 또는 BFS 를 적용하여 connected component 를 찾을 수 있다. Graph, DFS, BFS 는 자료구조 과목에서 배우는 내용이지만, 인터넷에서 찾을 수 있는 자료로부터도 쉽게 이해할 수 있을 것이다.