



[CS 11] Prac 6g – Really, Really Tall Flowers

Problem Statement

You have n flowers in your garden, each with a positive integer height. They form a single row. We label the flowers 1 to n from left to right, so flower 1 is the leftmost, while flower n is the rightmost.

Every day, you decided to water a *contiguous* portion of the flowers—that is, for some i and j such that $1 \leq i \leq j \leq n$, you watered flowers i to j , inclusive. Every time a flower is watered, its height increases by 2.

After d days, what are the heights of the flowers?

[Submit solution](#)

[CS 11]

Practice 6

[My submissions](#)

✓ Points: 250 (partial!)

⌚ Time limit: 6.0s

☰ Memory limit: 1G

✍ Author:

kvatienza (Kevin Atienza)

➤ Problem type

▼ Allowed languages

NONE, py3

Task Details

Your task is to implement a function called `flower_heights`. This function has two parameters.

- The first is a `tuple` of n `int`s denoting the initial heights of the flowers.
- The second is a `tuple` of d pairs of `int`s specifying the flowers you watered each day. Specifically, each pair looks like (i, j) , and it means you watered flowers i to j , inclusive, on that day.

The function must return a `list` of `int`s denoting the flower heights after the d days.

Restrictions

(See 6a for more restrictions)

For this problem:

- Loops and lists are allowed.
- Up to 8 function definitions are allowed.
- Recursion is **disallowed**. (The recursion limit has been greatly reduced.)
- Sets and dictionaries are **disallowed**.
- Generators and comprehensions are **disallowed**.
- The source code limit is 600.

Example Calls

Example 1 Function Call

```
flower_heights((3, 1, 4, 1, 5), (  
    (1, 4),  
    (3, 5),  
    (4, 4),  
)
```

[Copy](#)

Example 1 Return Value

```
[5, 3, 8, 7, 7]
```

[Copy](#)

Constraints

When your program is run:

- The function `flower_heights` will be called at most 50,000 times.
- The sum of n across all calls will be at most 150,000.
- The sum of d across all calls will be at most 150,000.
- $0 \leq n \leq 150,000$
- $0 \leq d \leq 150,000$
- $1 \leq i \leq j \leq n$ for each pair (i, j) in the second argument.
- Each height is a positive integer at most 10^{10} .

Scoring

- You get 50 ❤ points if you solve all test cases where:
 - $n \leq 50$
 - $d \leq 50$
 - the sum of n across all calls is at most 500.
 - the sum of d across all calls is at most 500.
- You get 100 ❤ points if you solve all test cases where:
 - $n \leq 4,000$
 - $d \leq 4,000$
 - the sum of n across all calls is at most 8,000.
 - the sum of d across all calls is at most 8,000.
- You get 100 ❤ points if you solve all test cases.

Clarifications

[Report an issue](#)

No clarifications have been made at this time.