Introduction to Analysis of Algorithms	Homework 8
CS 4820 Fall 2020	Due Tuesday, December 8, 2020

Instructions. Submit a .pdf file with your typeset solutions to questions 1 and 2 and a text file with your source code for question 3 in a .zip file to CMS. Do not include the compiled object or other compilation artifacts. You will also need to run your program on the autograder as with previous programming exercises.

All questions are worth 10 points. Remember that when asked to design an algorithm, you must also prove correctness and analyze its running time. When a problem asks for an efficient algorithm, you should strive to be as efficient as possible.

1. K&T Ch. 11 Ex. 1 p. 651.

2. K&T Ch. 11 Ex. 6 p. 654.

3. While randomly surfing the internet, you found out that if in Google you type "how were", one of the top suggestions is "how were the pyramids built." Being an expert algorithm designer, you decide to solve the problem yourself.

   Because building a 3-dimensional pyramid is hard, you decide to first solve the problem of creating a 2-dimensional pyramid. A 2-dimensional pyramid (hereafter simply called pyramid) consists of several identical squares. A pyramid of height h has the following structure:

   - There is a vertical stack of h squares at the center. Assume this stack is at horizontal position k.
   - Immediately left and right of the center, that is, at positions $k-1$ and $k+1$, there is a vertical stack $h-1$ squares high.
   - This structure repeats itself; that is, at position $k-i$ and $k+i$, there is a stack of $h-i$ squares, until positions $k-h$ and $k+h$, where there are no squares.

   In order to build a pyramid, you are given stacks of squares of various heights next to each other. However, you know that these squares are too heavy to move, so you cannot move a square from one stack to another; you can only delete squares (up to this day scientists cannot explain where these squares go after being deleted). Your goal is to find the height of the highest pyramid that can be built using this procedure, as well as where it is.

   Input format The input to your program will be provided as an ASCII character stream on the standard input. It will consist of two lines terminated by newline characters (`'\n'`, `0x0A`). The first line will contain a single integer n, the number of stacks of squares. The second line will contain n space-separated integers; the i-th integer represents the number of squares in the i-th stack. Assume that these stacks are right next to each other, in the order they are given.

   Output format Your program should output a single line terminated by a single newline containing two integers h and i. The number h should be the height of the highest pyramid that can be built by deleting squares from the initial configuration. The number i should be the index of the center/highest stack in the pyramid, where the index of the leftmost stack is 0. If multiple such i's exist, print the index of the leftmost one.

Example

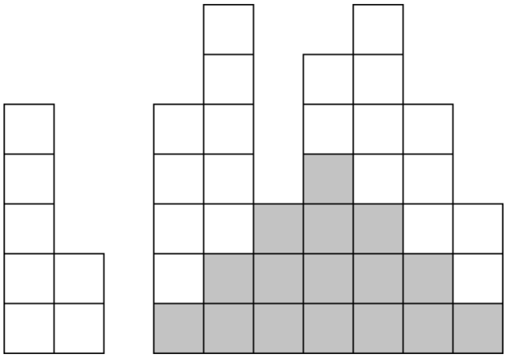
The input

```
10
5 2 0 5 7 3 6 7 5 3
```

should generate the output

```
4 6
```

The example is depicted on the right, with the gray squares depicting the highest pyramid.



Important points We guarantee that any input we provide to test your code will conform to the format described above, so you do not need to check for format errors. The number of stacks n will be at least 1. The height of each stack will be an integer in the range $[0, 10^9]$.

We will test your code on some large instances of the problem, so make sure your implementation is very efficient. The number of stacks n will most likely be in the range of 10^5.

You may use any of the following languages: Python, Java, C, C++, or OCaml. Please format your code attractively and provide copious comments. As before, we will grade your program using an autograder. The autograder can be accessed at https://cs4820.cs.cornell.edu/, code named *Pyramids*. The source code submitted to CMS must match the latest version uploaded to the autograder.