

Weekly Exercise #7

Dr. Moneybags

In this exercise, you will use what you have learned about sorting and runtime complexity to solve a simple problem. Although this problem is similar to a morning problem, it will be graded as a weekly exercise, which means that correctness, style, code design, and runtime complexity are all important in your final submission. Passing all tests is not enough to guarantee full marks on this exercise.

Important Notes:

- This is a Python exercise, run and compiled in the same way as a morning problem:
`./opentestcenter.sh`
- For this weekly exercise, **speed matters**. Your solution must run in $O(n \log(n))$ time to be accepted. To help you test your solution for speed, we have provided Test Center data. If you see a Timeout Error, your solution is not fast enough.
- **Hint:** If you are stuck on solving this problem, think about whether sorting or ordering the applicants' net worths might help.

Your Task: Dr. Moneybags' Prestigious Club

Dr. Moneybags wants to form an elite club composed of only the richest people in the world. We're not judging; they paid us a very large sum of money to help them form their club! With your expert CMPUT 274 knowledge, we have enlisted you to help as well.

Dr. Moneybags wants their club to be as large as possible, so it will become prestigious. However, they also want to maintain certain standards of wealth in their club, so that no single member of the club has fewer than $\$N$ million dollars. However, they are not sure what the best threshold N should be. They have decided that the best way to determine which members should be in their club is to do the following:

- Examine the net worth of each person applying to the club (Dr. Moneybags is a very famous and popular person, so there could be as many as 100000 applicants).
- Find the largest number N such that there are at least N applicants with at least N million dollars.
- Once you have found the value of N , accept anyone with at least N million dollars into the club.

Given an application pool where each applicant is represented by his or her net worth, can you determine the minimum wealth for each applicant given Dr. Moneybags's strange requirements?

Input

The first line of input contains a single integer n ($0 \leq n \leq 100000$), which is the number of applicants to the club.

The next n lines describe the applicants. Each of these lines contains a single integer w ($0 \leq w \leq 1000000000$), the net worth of a given applicant, in millions of dollars.

Output

Display your computed value for N , the minimum wealth threshold for applicants to the club, followed by a newline.

Sample Input 1

```
5
7
1
2
1
5
```

Sample Output 1

```
2
```

Explanation: Five people want to join the elite club. They have \$7, \$1, \$2, \$1, and \$5 million dollars, respectively. There are at least 2 applicants with at least \$2 million dollars (those with \$2, \$5, and \$7 million) but this is the largest value for N as there are not at least 3 people with at least \$3 million dollars. Therefore, Dr. Moneybags will only allow applicants with at least \$2 million dollars to enter their club.

Sample Input 2

```
5
7
1
3
1
5
```

Sample Output 2

3

Explanation: Again, there are five applicants. This problem is almost identical to the one above, except that the applicant with \$2 million dollars above is replaced with a different applicant who instead has \$3 million. This means that there are at least 3 people with at least \$3 million dollars, so the threshold N is 3.

Sample Input 3

```
3
40
20
30
```

Sample Output 3

```
3
```

Explanation: There are three applicants to the club, with \$40, \$20, and \$30 million dollars each. Dr. Moneybags can allow all three applicants to join their elite club, as there are at least 3 applicants with at least \$3 million dollars. Therefore, the minimum wealth threshold N of the club with this pool of applicants is 3.

Submission Guidelines:

Submit all of the required files (and no other files) as **one** properly formed compressed archive called either `moneybags.tar.gz`, or `moneybags.tgz`, or `moneybags.zip` (for full marks, please do **not** use `.rar`):

- when your archive is extracted, it should result in exactly *one directory* called `moneybags` (use this exact name) with the following files in that directory:
 - `moneybags.py` (use this exact name) contains all of your Python code and *docstrings-based documentation*.
 - your `README` (use this exact name) conforms with the Code Submission Guidelines.
 - No other files should be submitted.

Note that your files and functions must be named **exactly** as specified above.

A new tool has been developed by the TAs to help check and validate the format of your `tar` or `zip` file *prior* to submission.

To run it, you will need to download it into the VM, and place it in the same directory as your compressed archive (e.g., `moneybags.zip`).

You can read detailed instructions and more explanation about this new tool in Submission Validator: Instructions (at the top of the Weekly Exercises tab), or run:

```
python3 submission_validator.py --help
```

after you have downloaded the script to see abbreviated instructions printed to the terminal.

If your submission passes this validation process, and all validation instructions have been followed properly, you will not lose any marks related to the format of your submission. (Of course, marks can still be deducted for correctness, design, and style reasons, but not for submission correctness.)

When your marked assignment is returned to you, there is a 7-day window to request the reconsideration of any aspect of the mark. After the window, we will only change a mark if there is a clear mistake on our part (e.g., incorrect arithmetic, incorrect recording of the mark). At any time during the term, you can request additional feedback on your submission.