[CS 11] Prac 6a – Trace of a Matrix

Problem Statement

One definition of the **trace** is as follows. Given a linear map, its trace is the sum of its eigenvalues, counted up to multiplicity.

In terms of its matrix representation, the trace has a surprisingly simple formula. It is simply the sum of the main diagonal entries of the matrix! (The main diagonal is the one that goes from top-left to bottom-right.)

For example, the trace of the matrix $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ is $1 + 5 + 9 = 15$.

Indeed, the eigenvalues of the matrix are 0, $\frac{15+3\sqrt{33}}{2}$ and $\frac{15-3\sqrt{33}}{2}$, and their sum is indeed 15.

Because it is the sum of the eigenvalues, it is invariant under similarity, i.e., M and TMT^{-1} have the same trace. This can be surprising if the first definition of "trace" you encountered is the sum of the main diagonal entries—it's not entirely clear why M and TMT^{-1} would have the same sum of diagonal entries!

Given a matrix representing a linear map, compute its trace.

Task Details

Your task is to implement a function called `trace`. This function has a single parameter, a `tuple` of `tuple`s of `int`s representing a (square) matrix.

The function must return an `int` denoting the trace of the given matrix.

Restrictions

In this lab session, many names are banned. Here are a few of them: `zip`, `input`, `type`. This is *not* an exhaustive list. (If you accidentally use a variable name that turns out to be banned, please rename it.)

The following names are now allowed: `range`, `list`, `enumerate`, `print`, `append`, `pop`, `extend`, `remove`, `sort`, `insert`, `clear`, `reverse`, `reversed`.

This lab session focuses on loops. Therefore, recursion is banned. Also, generators and comprehensions are not yet allowed for this lab.

For this problem:

- Loops and lists are allowed.
- Up to 3 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

```
trace((
    (1, 2, 3),
    (4, 5, 6),
    (7, 8, 9),
))
```

Copy

Example 1 Return Value

```
15
```

Copy

Constraints

When your program is run:

- The function `trace` will be called at most 10 times.
- The matrix will be an $n \times n$ square matrix with $1 \leq n \leq 200$.
- Each element of the matrix will be an integer with absolute value at most 10^{10}.

Scoring

- You get 150 ❤️ points if you solve all test cases.

Clarifications

Report an issue

No clarifications have been made at this time.

Submit solution [CS 11] Practice 6 ❤️

My submissions ❤️

Points: 150 (partial)
Time limit: 6.0s
Memory limit: 1G

Author:
kvatienza (Kevin Atienza)

Problem type

Allowed languages
~~NONE~~, py3