

Recursive Average

Given a list of integers, calculate the average (arithmetic mean) of all elements using a **recursive approach**.

Important: Your solution **MUST** use recursion. Any non-recursive solution will receive zero points from the judge, regardless of correctness.

Recursive Approach: Break down the problem by:

- Base case: If the list is empty, return 0
- Recursive case: $\text{avg}(a) = \frac{(n-1) \times \text{avg}(a_{1:n-1}) + a_n}{n}$, where n is the length of the list a

Input Format

A list of integers $a = [a_1, a_2, \dots, a_n]$

Output Format

Return a single float representing the average of all elements

Constraints

- $1 \leq n \leq 1000$ (length of the list)
- $-10^6 \leq a_i \leq 10^6$ (each element)

Sample Input

```
a = [1, 2, 3, 4, 5]
```

Sample Output

```
3.0
```

Explanation

- Average of $[1, 2, 3, 4, 5]$ is $\frac{1+2+3+4+5}{5} = \frac{15}{5} = 3.0$

Implementation

Goal: Fill in the following function using **recursion only**:

```
def recur_avg(a: list[int]) -> float:
    ...
    return ...

if __name__ == "__main__":
    # You can test anything inside this block and can send it to grader
    # The grader will use only the function that you have implemented
    # !!! DO NOT write anything outside this block
    print(recur_avg([1, 2, 3, 4, 5]))
```

WARNING: The judge will automatically check if your solution uses recursion. Any non-recursive solution (using loops, built-in functions like `sum()`, etc.) will receive **zero points** regardless of correctness.