Recursion

Induction

- Reading: Why Induction?
 10 min
- Reading: What is Induction?
 10 min
- Reading: Arithmetic Series
 10 min
- Reading: Plane Coloring
 10 min
- Reading: Compound Interest
 10 min
- Lab: Bernoulli's Inequality
 15 min
- Reading: Inequality Between
 Arithmetic and Geometric Mean
 10 min
- Reading: More Induction Examples
 10 min
- Reading: Where to Start Induction?

 10 min
- Reading: Triangular Piece
 10 min
- Reading: Proving Stronger
 Statements May Be Easier!
 10 min
- Reading: What Can Go Wrong with Induction?

 10 min
- Quiz: Puzzle: Connect Points 2 questions
- Quiz: Induction
 9 questions

Why Induction?

You are reviewing your friend's python code. The code contains a function that, given a positive integer n, computes the sum of the first n positive integers: $1+2+\cdots+n$.

```
1 def sum_of_integers(n):
2   assert n > 0
3   return sum(range(1, n + 1))
```

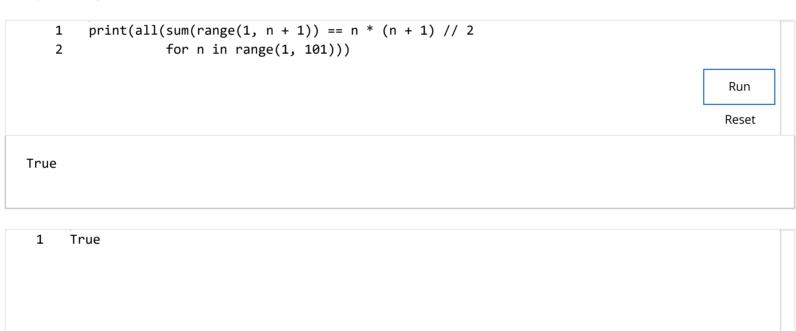
You suggest your friend to improve the code by using the *formula for arithmetic series*: for every positive integer n,

$$\sum_{i=1}^n i = 1 + 2 + \dots + n = rac{n(n+1)}{2}$$
 .

A code that uses a formula $\frac{n(n+1)}{2}$ instead of summing up all integers from 1 to n is more efficient in practice: already for $n=10^9$, computing the sum takes noticeable time, whereas computing $\frac{n(n+1)}{2}$ takes almost nothing.

How could you convince your friend that the formula $\frac{n(n+1)}{2}$ is true for *every* positive integer n?

One possibility is to check that it holds for all $1 \leq n \leq 100$.



This code snippet ensures that the formula above is true for all $1 \leq n \leq 100$.

But it only convinces us that the formula holds *sometimes*. Moreover, we have no possibility to check every positive integer n for a simple reason: there are infinitely many of them! Thus, to be sure that the formula holds for *all values of* n, we need a rigorous mathematical proof.

This is reminiscent of software testing. Whereas

one often can run a program on several tests, it is usually impossible to cover all cases with tests. This is why it is important to \emph{ prove } correctness of a program(for all input values).

Mathematical induction is a general way of proving statements of the form:

for all integer $n \geq c$, the statement A(n) is true.

Here, A(n) is a statement (that might be true of false) that depends on n. In our warm-up example above, A(n) asserts that $1+2+\cdots+n$ is equal to $\frac{n(n+1)}{2}$. In the next section, we introduce the method of mathematical induction and give many examples.

In general, checking a few first values of n to verify the correctness of A(n) is a bad idea. Later we'll encounter several examples where A(n) is true in the beginning, but eventually it becomes false. Moreover, there are examples of simple statements A(n) that are true for all $1 \le n \le 10^{80}$, but are false in general! That is, there exists n such that A(n) is false, but the minimum such n is so huge that it cannot be found by a brute force search.

✓ Comp	leted	Go to next item		
△ Like	□ Dislike	Report an issue		