Exercise 1-1  Collatz conjecture

For positive integers \( n \), \( f(n) \) is defined as

- \( \frac{n}{2} \) if \( n \) is even and
- \( 3 \cdot n + 1 \) if \( n \) is odd.

a) Implement the function \( f \) in Rust.

b) Add another function that uses \( f \) and that allows you to answer the following question: For a given \( n \), what is the smallest \( i \) such that \( (f^i)(n) = 1 \)? Answer this question for several values of \( n \), for example 12, 15, 27, 29.

c) Extend your program so that it also returns the highest intermediate value of \( (f^j)(n) \) for \( 0 \leq j \leq i \) and a given \( n \), i.e.

\[
\max(f^0(n) := n, f(f(n)), f(f(f(n))), \ldots, (f^{i-1})(n), (f^i)(n)).
\]

Give the highest intermediate values for all values of \( n \) that you used in the previous task.

d) Give the highest intermediate values as defined above for all \( n \in [1, 100] \).

Test your solution

Your solution is most likely correct if your results to subtasks b and c look like this:

for \( n = 12 \), smallest \( i \): 9, largest \( n \): 16
for \( n = 15 \), smallest \( i \): 17, largest \( n \): 160
for \( n = 27 \), smallest \( i \): 111, largest \( n \): 9232
for \( n = 29 \), smallest \( i \): 18, largest \( n \): 88

Hints

- In the first task you could use the function signature `fn f(n: u32) -> u32`.
- The function signature `fn find_smallest_i(n: u32) -> u32` would be a good start for the second task. This new function should call `f(n)` and keep track of the results in order to determine the smallest value \( i \).
- You can solve the exercise either recursively or iteratively. Due to its simplicity, using the recursive approach is highly suggested. If you choose the iterative approach you will have to take into account which variables/references need to be mutable. (Not yet covered in lecture, only for advanced students.)
- Extend the function written for the second task so that you are able to answer the remaining tasks.