

POETRY OF PROGRAMMING

CLOJURE PROJECTS

v2019.07.08

Solving these problems may require a bit of thinking and planning. This is exactly the challenge: decomposing the problem into simpler tasks.

(1) **The Caesar shift cipher**

The Caesar cipher is the simplest form of encryption, where each letter is substituted by another letter from the alphabet shifted by n letters. For example, "hello" can be encrypted as "ifmmp" when using the $n = 1$ shift cipher. Write functions that produce encrypter and decrypter functions for a given n . Write another function that performs a brute-force attack on the cipher by trying all possible shifts.

(2) **The halving method for finding roots**

The *root* of function is a value for x such that $f(x) = 0$. Write a CLJ function `find-root` that takes a continuous real function $f : \mathbb{R} \rightarrow \mathbb{R}$ and two real numbers a, b such that $f(a) < 0$ and $f(b) > 0$. This way f is bound to cross the x -axis at least once, and `find-root` can find a root by systematically halving the $[a, b]$ interval and calling itself recursively. It should work up to some predefined level of precision.

(3) **Efficient Collatz**

Calculate the return time of integers in the Collatz conjecture as efficiently as possible. This involves storing the return time for each intermediate number.

(4) **Maze solver** A maze is described by a string. Character # represents wall, . path, S the start point, and D the destination.

```
S...
###.
....
.#.#
...D
```

Write a program that outputs a path from start to destination. For instance, using o for the actual path taken.

```
Sooo
###o
..oo
.#o#
..oD
```