Map pattern

By using the map pattern, give a parallel solution for the following problems

Scaled vector addition

The *scaled vector addition* operation scales a vector **x** by a scalar value **a** and adds it to a vector **y**, elementwise. Both vectors have length n. This operation is frequently used in linear algebra operations, such as for row cancellation in Gaussian elimination. Suppose the ith element of x is x_i and the ith element of y is y_i. Then, the scaled vector addition operation is defined as follows:

$$f(t,p,q) = tp+q,$$
$$\forall i:\ y_i \leftarrow f(a,x_i,y_i)$$

Design and implement an algorithm to compute the scaled vector addition operation. Give the work and span of your algorithm. Finally, show experimentally the scalability of your implementation.

Mandelbrot set

The Mandelbrot set is the set of all points c in the complex plane that do not go to the infinity when the quadratic function $z \leftarrow z^2+c$ is iterated. In practice, it is hard to prove that this recurrence will never diverge so we iterate up to some maximum number of times. We can use a lookup table to map different counts to colors to generate an image.

$$z_0 = 0$$
$$z_{k-1} = z_k^2 + c,$$
$$\mathrm{count}(c) = \min(|z_k| \geq 2),\ 0 \leq k < K$$

We can find several implementations of the Mandelbrot set. In particular, see the file **mandelbrot.c**. Modify the implementation of the file **mandelbrot.c** to compute the Mandelbrot set in parallel.

Note: The parallel algorithm includes a **data-dependent control flow**. This leads to a load imbalance: Different pixels in the computation can take different numbers of iteration to diverge.

Breaking cryptography systems (naïve)

Consider the cipher function implemented in the file **cipher.c**. The function ciphers a sequence of digits (0 to 9) by mapping each digit to a different one, using a **reflector**. A reflector is a permutation of the alphabet 0 to 9. If we know the reflector, we can decipher any encrypted sequence. Assume that you have the cipher function, an encrypted sequence **<67330819503276251491449082604 8>** and you know that the original sequence starts with **<785>.** With the help of the map pattern:

- Find all possible reflectors which generated the encrypted sequence using a parallel brute force algorithm
- Improve your previous algorithm using the knowledge given in the description
- Now you have a new message **<B312409F8EE1DF1351463CF71D897>**, but now over the hexadecimal alphabet. If you know that all the hexadecimal sequences start with **<A0>**, find all possible reflectors.

String matching (naïve)

Let $\sum$ be an finite alphabet. The string matching problem is defined as follows: Let T and P two sequences over $\sum$; find all the occurrences of P in T. In general, T is called the *text* and P is called the *pattern*. Following the map pattern, implement a parallel algorithm to solve the string matching problem.