

Operator Problem

Write a program that outputs all possibilities to put the operators '+', '-', or nothing between the numbers 1, 2,..., 9 (in this order) such that the result is 100. For example $1 + 2 + 3 - 4 + 5 + 6 + 78 + 9 = 100$.

Solution

This solution was tested on Ruby 2.4.3 but should run on earlier versions. The `OperatorAlgorithms` class in `operator_algorithms.rb` contains all the algorithm functionality. The entry point for the algorithm is `find_operators` method. It calls `find_operators_recursive` with the initial empty `existing_operators` array. This is a divide and conquer algorithm. For each possible operator, the recursive function creates the next operator, and calls itself with that operator. The base case happens when there is an operator assigned to each spot between the numbers (including no operator). The algorithm uses several helper methods to calculate the sum with the assigned operators. This is returned up the chain such that the initial call receives a list of all the combinations of operators that provide a solution.

The `value_with_operators` method calculates the sum of any set of numbers and operators by using `merge_operators_with_numbers` to have an array with all the digits and operators in order. It iterates over this entire array. It keeps a buffer of digits and adds each consecutive digit it encounters to the buffer. When it reaches an operator, or the end of the array, the contents of the buffer is taken along with the previously seen operator to perform the operation on the sum. The operation and digit conversion are handled by the `do_operation` and `num_arr_to_int` methods.

Running the Examples and Unit Tests

`run_example.rb` contains an example of the algorithms at work. You can run it with:

```
$ ruby run_example.rb
```

`test_binary_tree.rb` contains unit tests for some of the algorithms. You can run them with:

```
$ ruby test_operator_algorithms.rb
```