Detailed Explanation of C++ Code for Roman Numeral Conversion

Diana Akolzina

December 29, 2023

1 Introduction

This document provides an in-depth explanation of the C++ code for converting Roman numerals to integers and vice versa, focusing on the specific features of C++, data structures, and algorithms used.

2 The RomanNumerals Class

The code defines a class named `RomanNumerals` in C++, encapsulating the functionality for Roman numeral conversion.

2.1 Class Members

The class contains two private member variables: `romanToIntMap` and `intToRomanMap`, both used for the conversion process.

```
map<char, int> romanToIntMap = {
    {'I', 1}, {'V', 5}, {'X', 10}, {'L', 50}, {'C', 100}, {'D', 500}, {'M', 1000
};
vector<pair<int, string>> intToRomanMap = {
    {1000, "M"}, {900, "CM"}, {500, "D"}, {400, "CD"},
    {100, "C"}, {90, "XC"}, {50, "L"}, {40, "XL"},
    {10, "X"}, {9, "IX"}, {5, "V"}, {4, "IV"}, {1, "I"}
};
```

Detailed Explanation:

- `romanToIntMap`: A map from `char` to `int` for mapping Roman numeral characters to integers.
- `intToRomanMap`: A vector of pairs, where each pair consists of an integer and its corresponding Roman numeral string.
- These data structures provide fast access and efficient storage for conversion values.

2.2 Method: romanToInt

Converts a Roman numeral string to an integer.

```
int romanToInt(string s) {
    int total = 0;
    int prev_value = 0;
    for (int i = s.length() - 1; i >= 0; i--) {
        int value = romanToIntMap[s[i]];
        if (value < prev_value)
            total -= value;
        else
            total += value;
        prev_value = value;
    }
    return total;
}
```

Detailed Explanation:

1. Initialize `total` and `prev_value` to 0.
2. Iterate over the string `s` in reverse order.
3. For each character `char` in `s`, retrieve its corresponding integer value from `romanToIntMap`.
4. Apply the Roman numeral conversion logic: subtract the value if it's smaller than `prev_value`, otherwise, add it.
5. Update `prev_value` for each iteration.
6. Return the final `total`.

2.3 Method: intToRoman

Converts an integer to a Roman numeral string.

```
string intToRoman(int num) {
    string roman;
    for (auto& pair : intToRomanMap) {
        while (num >= pair.first) {
            roman += pair.second;
            num -= pair.first;
        }
    }
    return roman;
}
```

Detailed Explanation:

1. Initialize a string `roman` to build the Roman numeral.
2. Iterate over `intToRomanMap` using a range-based for loop.
3. In each iteration, check if `num` is greater than or equal to the pair's first value.
4. If true, append the pair's second value (Roman numeral) to `roman` and decrement `num` by the pair's first value.
5. Repeat this process until `num` is less than the current pair's first value.
6. Return the final `roman` string.

3 Main Function

The main function demonstrates the usage of the `RomanNumerals` class.

```
int main() {
    RomanNumerals converter;
    cout << converter.romanToInt("MCMXCIV") << endl; // Output: 1994
    cout << converter.intToRoman(1994) << endl;      // Output: MCMXCIV
    return 0;
}
```

Explanation:

- Create an instance of `RomanNumerals`.
- Call `romanToInt` with "MCMXCIV" and display the output.
- Call `intToRoman` with 1994 and display the output.