CSE250 Project #4

Sheng Liu

July 2021

1 Introduction

In this project, you will implement a decoding function that decodes Huffman codes into characters. Huffman code is a kind of variable-length code that can be used for data compression. It is still used in ZIP files today, though there are better compression algorithms. In practice, Huffman code is "implemented" / "obtained" from a Huffman tree. Lecture introducing Huffman trees and how Huffman code can be decoded has been uploaded to:
https://drive.google.com/file/d/1cGk8GhpkKUkXmyqpB9Bn34WPjX8BD1Jc/view?usp=sharing

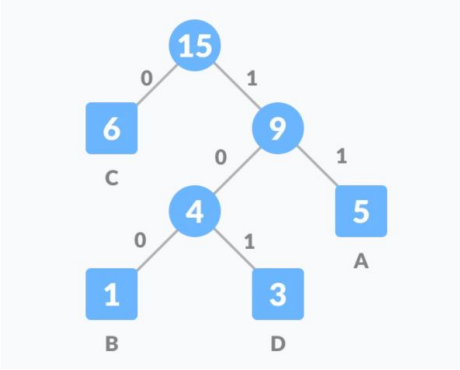

Figure 1: A Huffman tree that encodes "`A`", "`B`", "`C`", "`D`".

2 Functions and Assumptions

As promised, project 4 is relative simple. You will only implement one function, and there is no limitation on using functions defined in the `std` namespace. You will be able to implement this function with less than 30 lines of code.

- `string decode(Node *root, string huffmanCode)`: takes as input a pointer to the root node of a Huffman tree and a string containing a Huffman code, *i.e.*, a sequence of 0s and 1s. It returns the decoded string.

For example, given a Huffman tree shown in Figure 1 and "`10001011111`", it returns "`BCDAA`".

You can make the following assumptions:

- The given Huffman tree is a valid Huffman tree and `root` is not a null pointer, *i.e.*, `nullptr`.
- The given Huffman tree only encodes English letters, comma and the 10 digits. Other weird characters will not be encoded by the given Huffman tree.
- The given Huffman code is valid. In other words, it can be decoded with the given Huffman tree.

3 Submission

You **only** need to submit the hpp file named "decoding.cpp" to AutoLab. Do **NOT** modify the code we provided in the cpp file.