

Maitri Radhesh
CSU ID# 2831749

What is Huffman Coding?

Huffman coding is a lossless data compression algorithm. The idea is to assign variable-length codes to input characters, lengths of the assigned codes are based on the frequencies of corresponding characters.

The variable-length codes assigned to input characters are Prefix Codes, means the codes (bit sequences) are assigned in such a way that the code assigned to one character is not the prefix of code assigned to any other character. This is how Huffman Coding makes sure that there is no ambiguity when decoding the generated bitstream.

There are mainly two major parts in Huffman Coding:

1. Build a Huffman Tree from input characters.
2. Traverse the Huffman Tree and assign codes to characters.

Algorithm:

The method which is used to construct optimal prefix code is called Huffman coding.

This algorithm builds a tree in bottom-up manner. We can denote this tree by T

Let, $|c|$ be number of leaves

$|c| - 1$ are number of operations required to merge the nodes. Q be the priority queue which can be used while constructing binary heap

In my code, I am using the **Heap** data structure in order to analyze the tree of the "Huffman Code". It makes it easier to iterate it if I am considering it as a heap as I can move towards the right-sub-tree and left-sub-tree easily.

What is a Heap data structure?

A Heap is a special Tree-based data structure in which the tree is a complete binary tree.

Operations of Heap Data Structure:

- Heapify: a process of creating a heap from an array.
- Insertion: process to insert an element in existing heap time complexity $O(\log N)$.
- Deletion: deleting the top element of the heap or the highest priority element, and then organizing the heap and returning the element with time complexity $O(\log N)$.
- Peek: to check or find the most prior element in the heap, (max or min element for max and min heap).

Types of Heap Data Structure:

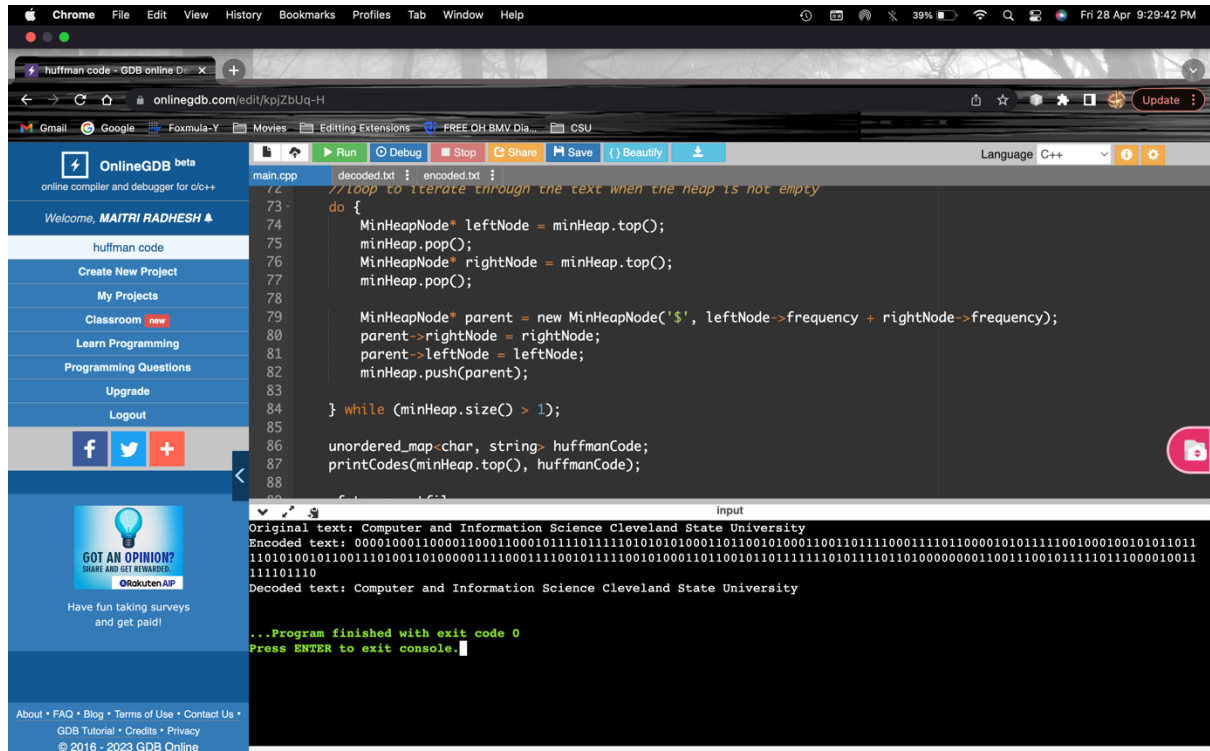
Generally, Heaps can be of two types:

1. Max-Heap: In a Max-Heap the key present at the root node must be greatest among the keys present at all of its children. The same property must be recursively true for all sub-trees in that Binary Tree.
2. Min-Heap: In a Min-Heap the key present at the root node must be minimum among the keys present at all of its children. The same property must be recursively true for all sub-trees in that Binary Tree.

I am making use of the a **MinHeap**.

Outputs:

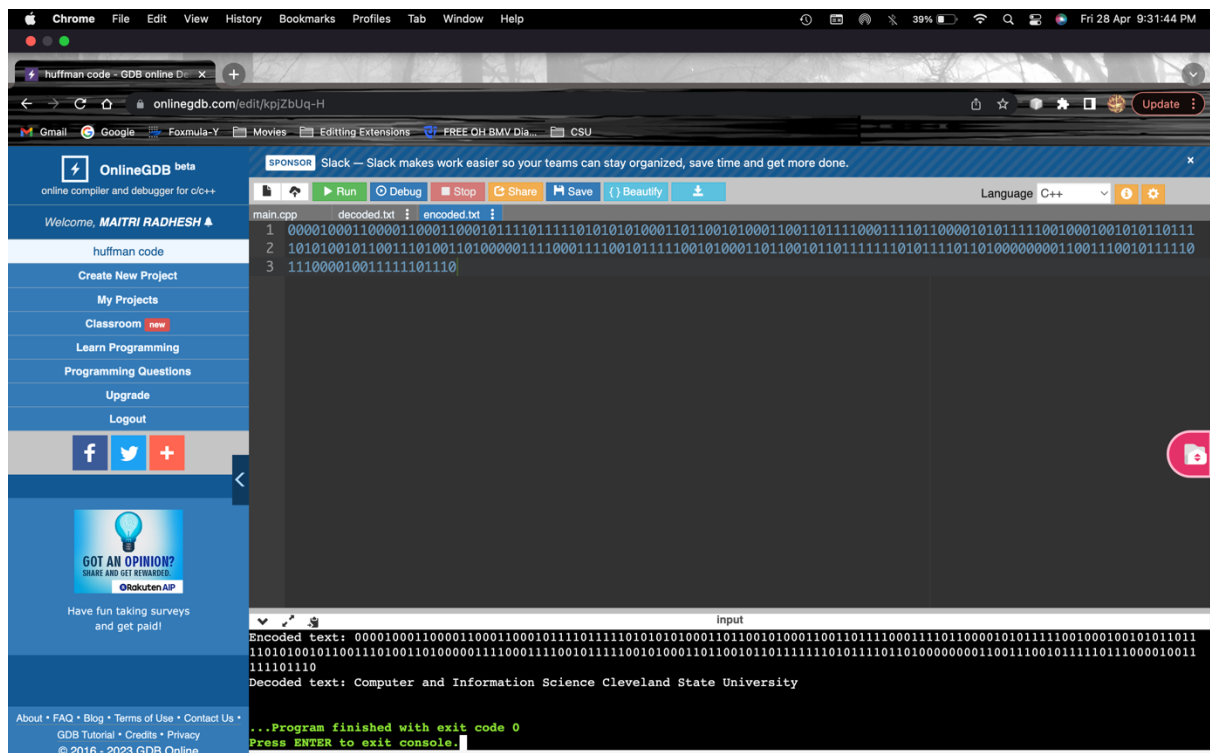
On the console:



The screenshot shows the OnlineGDB interface with a C++ program for Huffman encoding. The program uses a MinHeap to build a Huffman tree. The console output displays the original text, the encoded text as a long binary string, and the decoded text, which matches the original. The program finishes with exit code 0.

```
main.cpp | decoded.txt | encoded.txt |
72 //loop to iterate through the text when the heap is not empty
73 do {
74     MinHeapNode* leftNode = minHeap.top();
75     minHeap.pop();
76     MinHeapNode* rightNode = minHeap.top();
77     minHeap.pop();
78
79     MinHeapNode* parent = new MinHeapNode('$', leftNode->frequency + rightNode->frequency);
80     parent->rightNode = rightNode;
81     parent->leftNode = leftNode;
82     minHeap.push(parent);
83
84 } while (minHeap.size() > 1);
85
86 unordered_map<char, string> huffmanCode;
87 printCodes(minHeap.top(), huffmanCode);
88
Input
Original text: Computer and Information Science Cleveland State University
Encoded text: 00001000110000110001100010111101111010101000110110010100011001101110001111011000010101111001000100101011011
11010100101001110100110100000111000111100101111001010001101100101111101011101010000000110011100101111011000010011
111101110
Decoded text: Computer and Information Science Cleveland State University
...Program finished with exit code 0
Press ENTER to exit console.
```

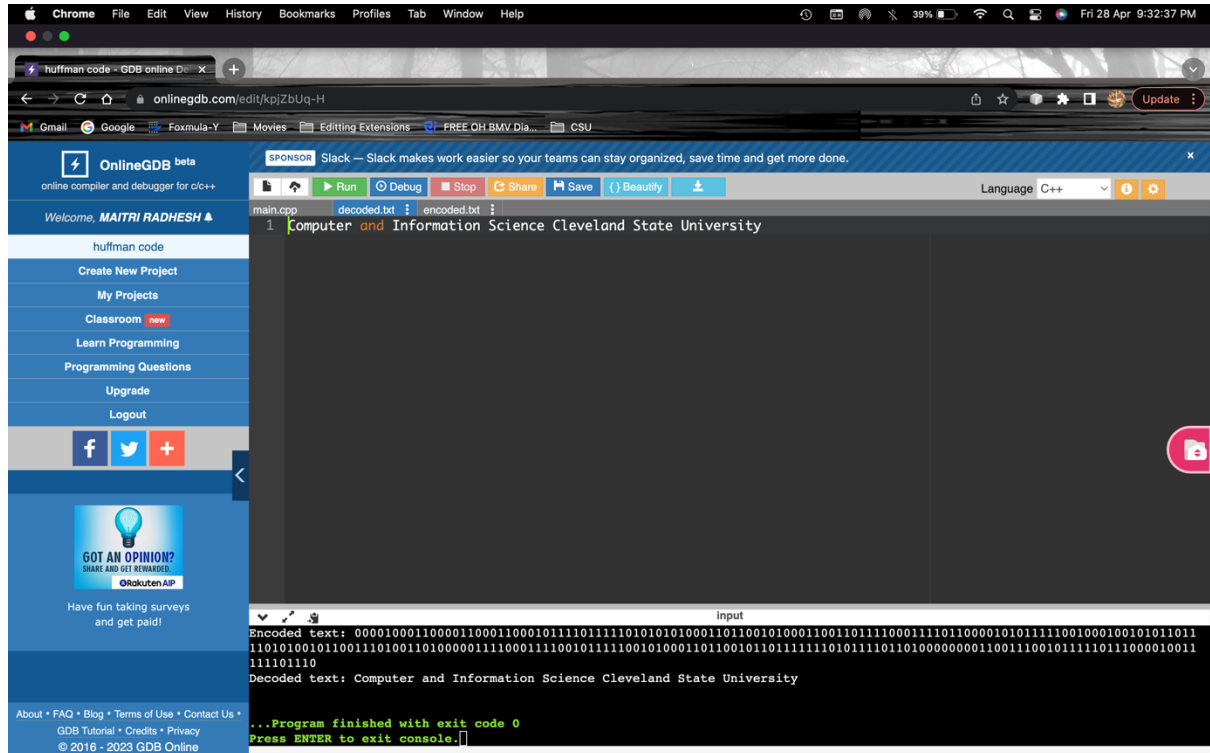
Encoded.txt:



This screenshot shows the same OnlineGDB interface, but the 'encoded.txt' tab is selected. It displays the encoded text as a long binary string, which is the result of the Huffman encoding process. The console output at the bottom shows the decoded text and the program's exit status.

```
main.cpp | decoded.txt | encoded.txt |
1 00001000110000110001100010111101111010101000110110010100011001101110001111011000010101111001000100101011011
2 101010010110011101001101000001110001111001011110010100011011001011111010111101101000000011001110010111110
3 111000010011111101110
Input
Encoded text: 00001000110000110001100010111101111010101000110110010100011001101110001111011000010101111001000100101011011
11010100101001110100110100000111000111100101111001010001101100101111101011101010000000110011100101111011000010011
111101110
Decoded text: Computer and Information Science Cleveland State University
...Program finished with exit code 0
Press ENTER to exit console.
```

Decoded.txt:



The screenshot shows the OnlineGDB website interface. The browser address bar displays `onlinegdb.com/edit/kpjZbUq-H`. The left sidebar contains navigation links: "huffman code", "Create New Project", "My Projects", "Classroom", "Learn Programming", "Programming Questions", "Upgrade", and "Logout". Below these are social media icons for Facebook, Twitter, and a generic share icon. A "GOT AN OPINION?" survey banner from Rakuten AIP is also visible. The main editor area shows a C++ file named `main.cpp` with the following code:

```
1 Computer and Information Science Cleveland State University
```

The "Run" button is highlighted. The output console at the bottom shows the decoded text and the program's exit status:

```
input
Encoded text: 0000100011000011000110001011110111110101010001101100110011110001111011000010101111001000100101011
11010100101100111010011010000011110001111001011110010100011011001011111110101110110100000001100111001011110111000010011
111101110
Decoded text: Computer and Information Science Cleveland State University
...Program finished with exit code 0
Press ENTER to exit console.
```

I have made use of an online compiler.

Link: <https://www.onlinegdb.com/edit/kpjZbUq-H>