Project Title: Huffman Coding Compression and Decompression

Description: Huffman coding is a popular algorithm used for lossless data compression. This project provides a Python implementation of Huffman coding for compressing and decompressing text files. The Huffman coding algorithm works by assigning variable-length codes to input characters, with shorter codes assigned to more frequent characters, resulting in efficient compression.

HuffmanCoding Class:

* This class encapsulates the functionality for Huffman coding compression and decompression.
* It includes methods for compressing and decompressing files, as well as helper methods for constructing Huffman trees and generating Huffman codes.

Huffman Tree Construction:

* The project constructs a Huffman tree based on the frequency of characters in the input text.
* It uses a priority queue (heap) to efficiently merge nodes with the lowest frequencies into a single tree.

Compression:

* During compression, the project reads the input text file, constructs a Huffman tree, and generates Huffman codes for each character.
* It encodes the input text using the generated Huffman codes and pads the encoded text to ensure a multiple of 8 bits for byte alignment.
* Finally, it writes the compressed data to an output binary file.

Decompression:

* During decompression, the project reads the compressed binary file and reconstructs the Huffman tree.
* It removes padding from the encoded text, decodes the text using the Huffman tree, and writes the decompressed text to an output text file.
* Usage:

To compress a text file, the user provides the path to the file.

* The program then compresses the file using Huffman coding and outputs a binary file with the ".bin" extension.
* To decompress a compressed binary file, the user provides the path to the compressed file.
* The program decompresses the file and outputs a text file with the "\_decompressed.txt" extension.
* Benefits:
* Huffman coding offers efficient compression for text data, especially when certain characters occur more frequently than others.
* The project provides a simple yet effective implementation of Huffman coding in Python.
* It can be used to compress large text files, saving storage space and reducing transmission times.