Programming Project

Project Title:

File Compressor Using C Programming.

Project Members :

S. M. Shaiful Alam (19)

Md. Saiful Islam (29)

Project Overview

Sometimes file takes some extra spaces unnecessarily even though it doesn't need at all. For that reason we have to carry more size of that file and it wastes memory of our hard-disk or any other memory drives. So we are going to write a program that will compress size of any type of files i.e. text file, image file, video file etc so that we can save memory spaces in some extent.

Related Work

Lessening memory is one of the most important issue of computer science. So far computer scientist are working regarding this vital issue. For that reason we've already found some work on file compressing using different algorithms in different times. Zip file is one of that which compresses different types of file to reduce memory and decompresses when needed to be executed.

Software Requirement

1. Windows Operating System
2. Integrated Development Environment (IDE) for C language i.e. Microsoft Visual C++, CodeBlocks etc.

Development Process

So far we've seen that many algorithm and development process have been implemented in this regard. Most used and famous algorithm is Huffman Algorithm. We are also going to implement this algorithm in C language. The entire development process are briefly described here…

Huffman coding

Huffman coding assigns shorter codes to symbols that occur more frequently and longer codes to those that occur less frequently. For example, imagine we have a text file that uses only five characters (A, B, C, D, E). Before we can assign bit patterns to each character, we assign each character a weight based on its frequency of use. In this example, assume that the frequency of the characters is as shown in Table 1.

Table 1 Frequency of characters

Character	A	B	C	D	E
Frequency	17	12	12	27	32

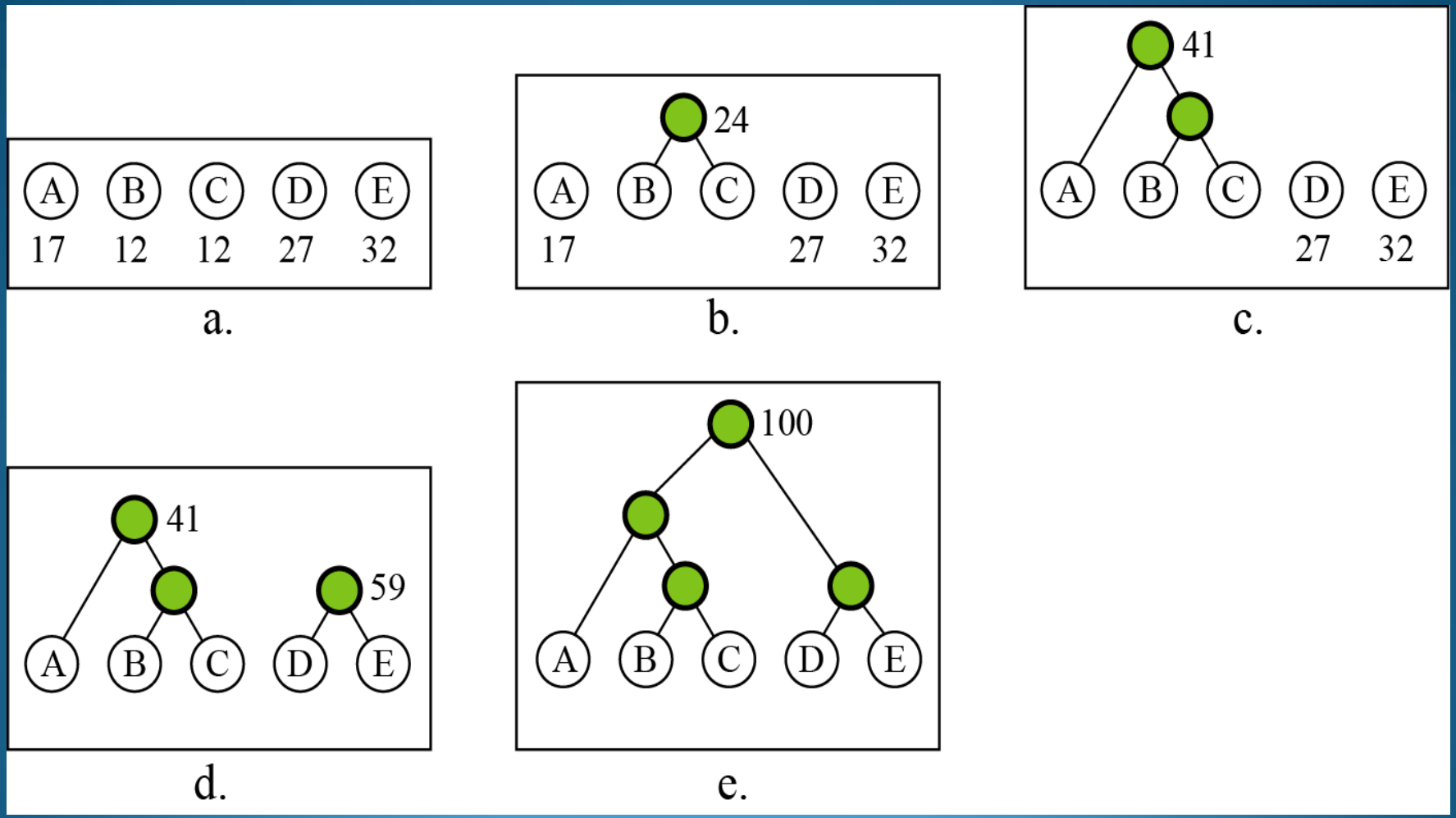


Figure 1 Huffman coding

A character's code is found by starting at the root and following the branches that lead to that character. The code itself is the bit value of each branch on the path, taken in sequence

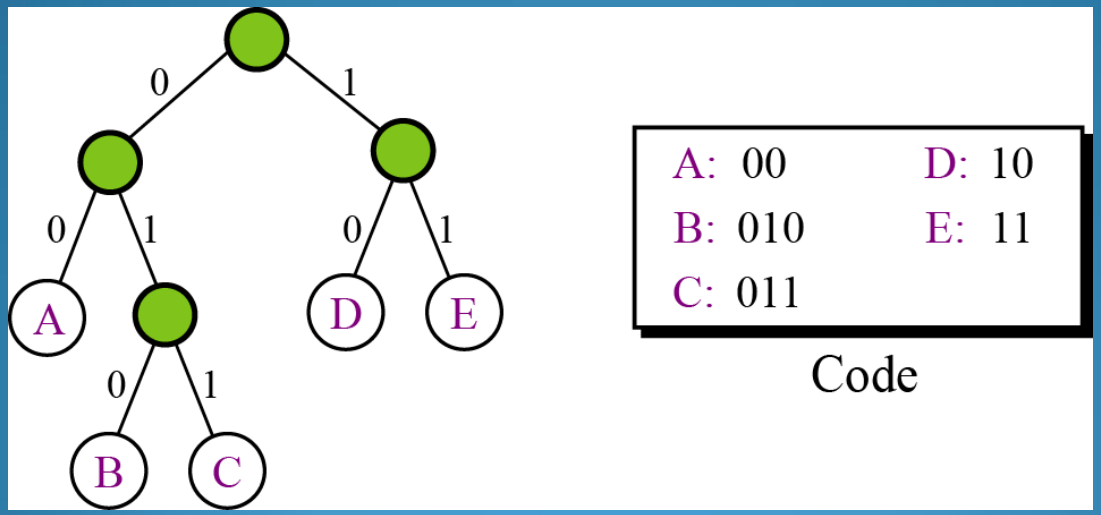


Figure 2 Final tree and code

Encoding

Let us see how to encode text using the code for our five characters. Figure 3 shows the original and the encoded text.

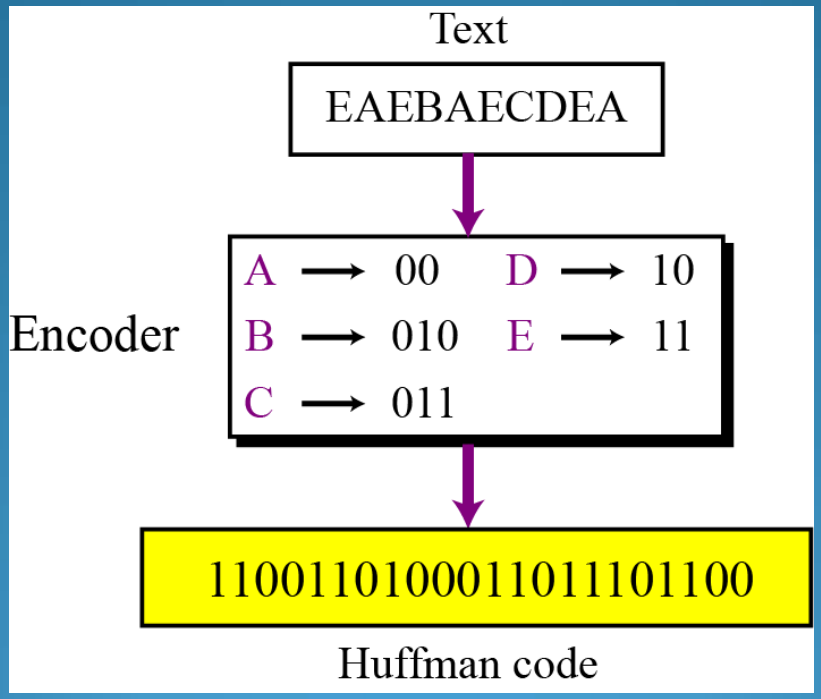


Figure 3 Huffman encoding

Decoding

The recipient has a very easy job in decoding the data it receives. Figure 4 shows how decoding takes place.

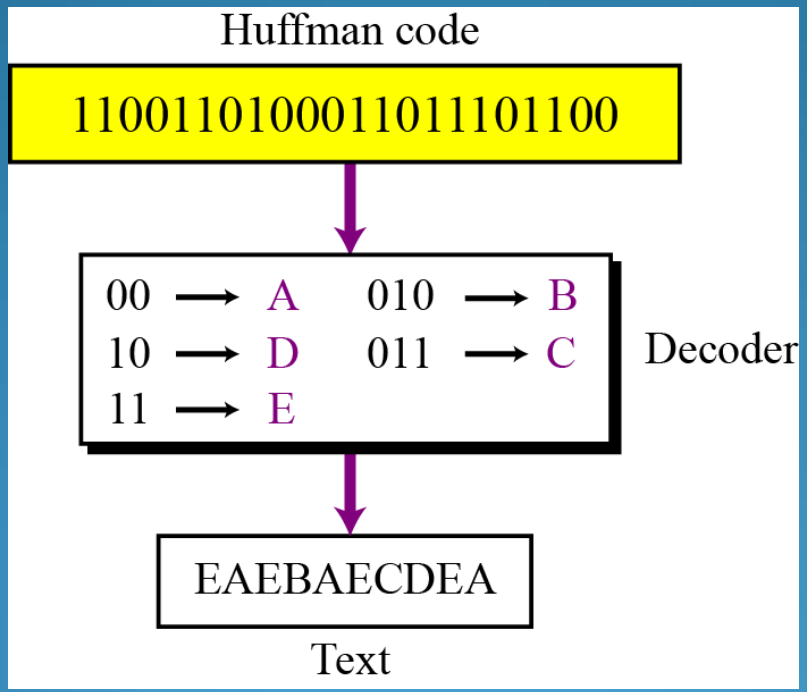
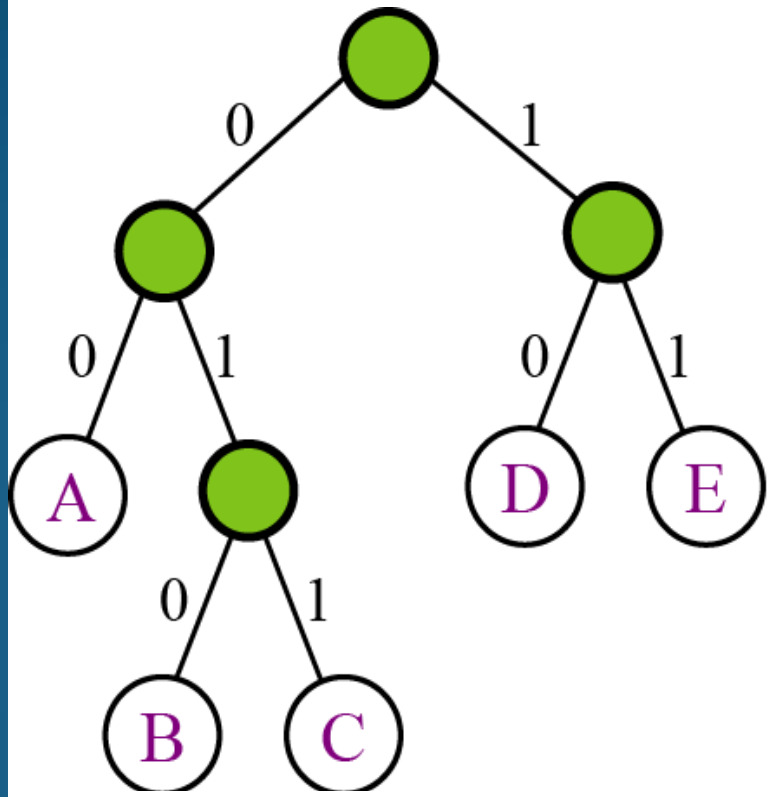


Figure Huffman decoding

Illustration:

1100110100011011101100



A: 00	D: 10
B: 010	E: 11
C: 011	

Code

EAEBAECDEA

Text

Result

We have implemented the algorithm using C programming and successfully compressed and decompressed text file….

Result Analysis

Here, we have a text file named sample.txt….

- Size of main file : 49427 bytes
- Size of compressed file : 28886 bytes
- Reduced size : 20541 bytes
- Percentage of the saving memory : 41.558%

Conclusion

We have reduced memory about 41% percent. So we think, we are in the track as our plan. If we can modify our thoughts in different part of our code, we hope we can reduce size more than present….

That's all from us....

Thank you