**University of Engineering and Technology**

**(Lahore campus)**

**Name:** Shahber Fatima (2018-CS-53)

Sana Aslam (2018-CS-68)

**Subject:** Analysis of Algorithm

**Submitted To:** Sir Samyan

**Project Title:** File Compression Tool

**PROJECT REPORT**

**File Compression Tool**

**Introduction:**

File compression algorithm is used to reduce storage requirement such as hard disk space .The main objective of this project is to achieve faster file transfer .Data or file compression is an art of representing data in compact form .file compression is a coding process that used number of bits to encode information or data. The main principal of file compression is to transfer string of character into new string which contain same information but with a new length as small as possible

For compression of data we used “**HUFFMAN CODING”**. Huffman coding used for data compression. In this algorithm, a variable-length code is assigned to input different characters. The code length is related to how frequently characters are used. Most frequent characters have the smallest codes and longer codes for least frequent characters. Letter with high frequency assign smallest code and letter with low frequency assign longest code .Every letter is assigned a binary string either ‘0’ or ‘1’.Huffman coding is greedy algorithm that used greedy algorithm in its implementation. Greedy algorithm is more efficient than dynamic programming .Greedy algorithm is faster than dynamic programming .Greedy algorithm is subset of dynamic programming .dynamic programming is slow .in greedy algorithm we make a choice and never look at back and hope that our solution is optimal and it gives minimum cost .Main concept of greedy algorithm is that we divide problem into sub problems and for each step we take best choice and hope that globally optimal solution

Huffman coding used for file compression

1. .Most frequent letter has smallest code and longer code for least frequent letter. Huffman tree consist of two parts:

* First we create Huffman tree
* Traverse tree to find code

**COST=** **Frequency \* no of bits of code**

1. Pre-fix free coding

* Any code cannot be prefix of another code

Example if character is assigned a ‘0’ code then other character will not be assig code like ‘01’ because it became prefix

* Every character is assigned a binary string it may be ‘0’ or ‘1’

1. No encoded string is prefix of any other
2. More frequent letter get smallest code
3. A prefix free code is a tree in which every tree has its own unique path in such a way no conflicts occurs. Tree which has minimum cost is good Huffman tree .greedy sub tree grow from bottom up .node with smallest frequency are combine and add their frequency into another node in this way number of nodes will reduce

**∑=depth(x) .p(x)**

Depth is length and p(x) is the probability

**END USER:**

In file compression, user can do following things:-

1. Selection of file
2. Compression of file
3. Decompression of file
4. Encode of file
5. Decode of file
6. Merge of file

In file compression user can select file which need to compress after selection of file, file is compressed or decompressed. User can encode or decode file or may merge two file

**Input format:**

Input format of file compression is that which file is selected for compression of data .in graphical user interface we use list view to show all folder in required directory .and there we show path in text box and button of selection where we can select ‘txt’ file .when selection button is clicked, there comes a list of folder from where selection of file is done .at the time when selection of file is done in text box there show a path of selected file

**Output format:**

Output format of file compression is that after file selection we go to next graphical user interface which is compression of file ,in this there is button of compression when clicked on it selected file is compressed .in compression button following functionality is added

* **PrintFrequency**

This function print frequency or probability of character. Every character is given frequency. .Most frequent letter has smallest code and longer code for least frequent

* **BuildTree**

This function build Huffman tree by adding least frequency of two node in this way two node combine and give one node and so no of nodes will reduce

* **SetHuffcode**

When this function is called every character is given a specific code .letter with high frequency has smallest code and letter with small frequency has longer code

* **Ascii**

 The ASCII function converts a string in EBCDIC code into ASCII code. An expression evaluating to the string to be converted. The ASCII function converts each character of the given expression from its EBCDIC representation value to its ASCII representation value.

* **FileWrite**

A file can be open in read or write mode we can write to a file after creating its name, by using the function FileWrite

* **PrintCode**

This function converts code begins edited into an html file display it by browser and print it

* **HuffTree**

This function made huff tree and give minimum cost by adding nodes who has least frequency

* **compressFile**

This function compress file with same information but with small length as possible as

* **Extension**

This function give extension of compressed file

**Pseudo code:**

**Algorithm Huffman(X)**

**Input:** String x of length n with d distinct characters

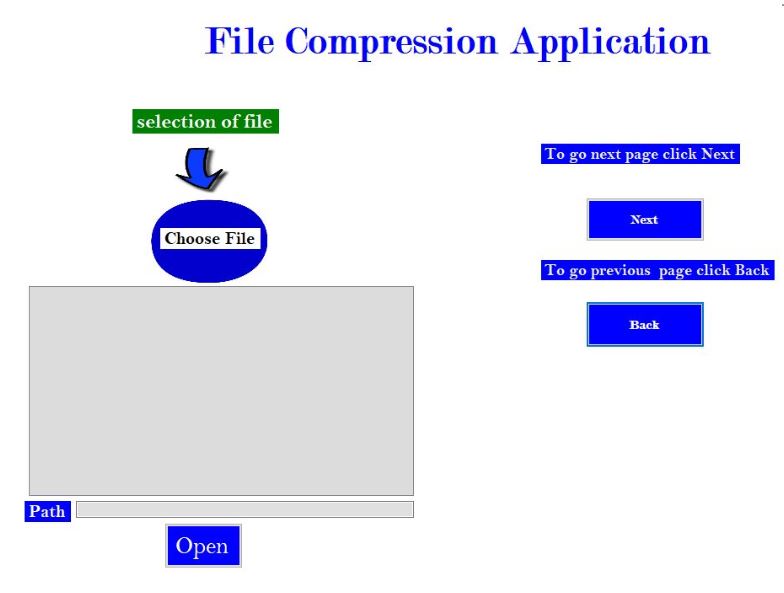
**Output:** coding tree for X

1. Compute the frequency f (c) of each character c of X.
2. (c1, f[c1] ) , (c2, f[c2] ) ,…, (cn, f[cn] )
3. Initialize a priority queue Q
4. For i=1 to n-1 Do
5. Create a single-node binary tree storing c
6. Insert T into Q with key f (c).
7. While Size ( )> 1 do
8. *F1* Q.min()
9. *T1* Q.removeMin()
10. *F2* Q.min()
11. *T2* Q.removeMin()
12. Create a new binary tree T with left sub tree T1 and right sub tree T2
13. T.left=T1 T.right=T2 Insert T into Q with key f1 +f2
14. T.f = T1.f1 + T2.f2
15. return Tree Q.removeMin()

**Interface:**

Following are the graphical user interface for file compression:-

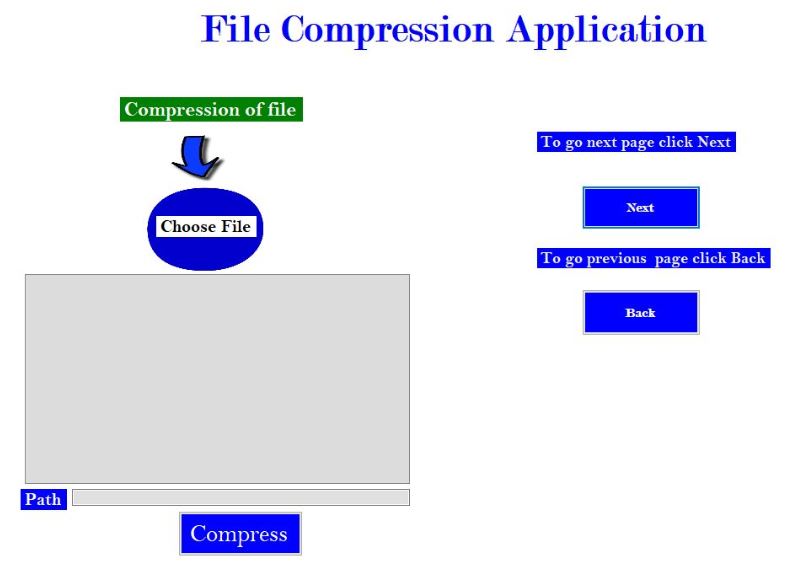
* Selection of file
* Compression of file
* Decompression of file
* Encoded of file
* Decoded of file



**Selection of interface:**

**Description:**

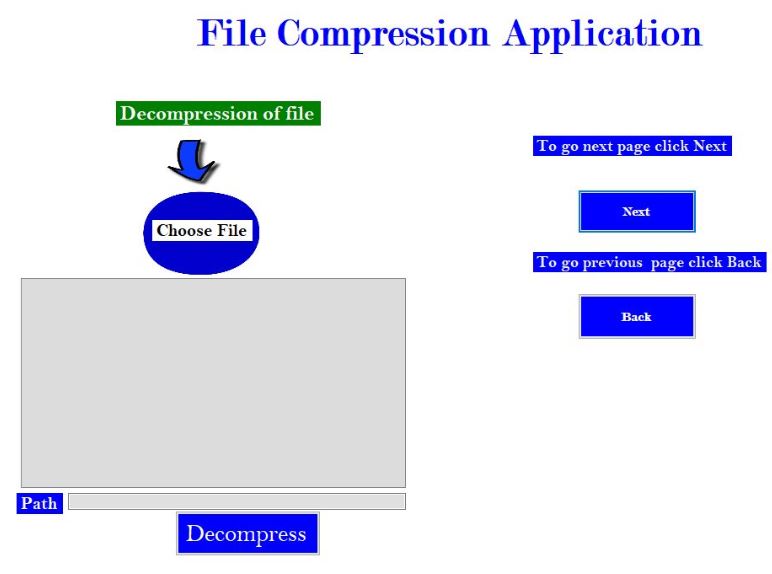
In this interface we select file which need to be compressed by clicking on button open there will open all folder then select txt file .in text box there show path of selected file



**Compression of file:**

**Description:**

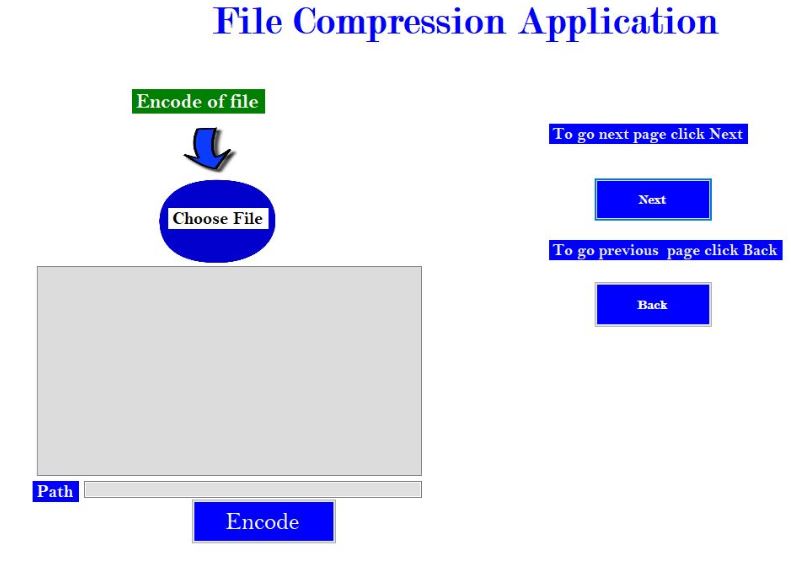
In this interface we compress the selected file by clicking on compress button .in compress button many functionalities has been done like printfrequency, BuildTree, setHuffcode, ASCII, FileWrite, PrintCode, HuffTree, CompressFile or Extension



**Decompression of file:**

**Description:**

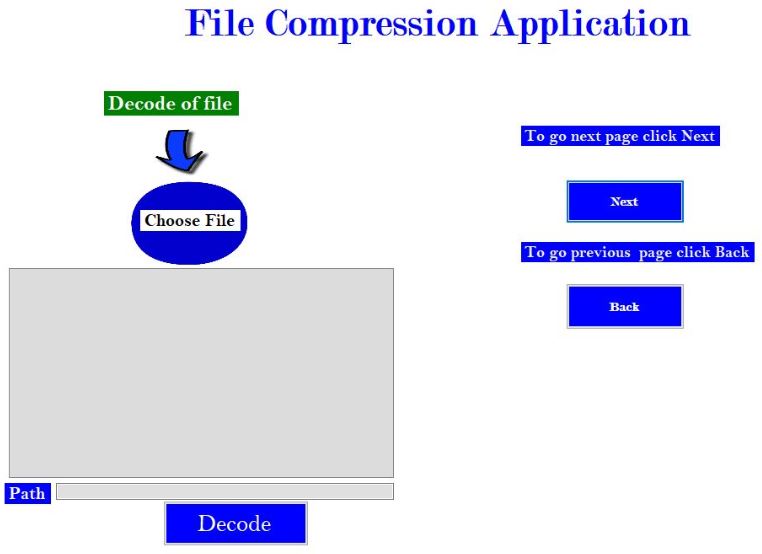
In this interface after selection of file, compression of file is done now compressed file is decompress by clicking on decompress button



**Encode of file:**

**Description:**

In this interface file is encode by clicking on encode button encoding is a process of converting data from one form to other form .by encoding the file it can be save into more efficient compressed format.



**Decode of file:**

**Description:**

In this interface file is decode by clicking on decode button. Decoded is opposite process it is conversion of encoded file into original file or sequence of character