Huffman Archiver

SDP 2016

Archiver class (archiver.h)

Private Members:

map<char, string>* mapTable;
histogram* freqTable; - Frequency table

vector<char> data; - vector of the bit data
tree* huffTree; - The huffman tree
int size; - Size of the compressed array in bits
int oSize; - Original size of the input string

Public Methods:

archive(string input = "") - Constructor by input string

void exportArchive(const char* dstName) - Export archive object to archive file

void importArchive(const char* dstName) - Parse archive file to archive object

void exportExtract(const char* dstName) - Decompress and extract to destination file

bool compressFile(const char* fileName) - Compress file using the compress function

string decompress();

void compress(string& input) - Compress the input string (using Huffman tree)

double compressionRate() - Return the compression rate

void printBits() - Prints the bits of the compressed string

void printNums() - Prints the bits of the compressed string as integers

void printMap() - Prints the replacement map

void printHist() - Prints the frequency histogram

void printTree() - Prints the Huffman tree

Tree Class (tree.h)

tree(**int** weight, tree* left, tree* right) - Construct by weight and left and right children

tree(**int** weight = 0) - Construct by weight

tree(**int** value, **int** weight) - Construct by value and weight

tree(node& n) - Construct by root node

void print() - Print the tree

node* getRoot() - Return pointer to the root

bool leaf() - Check if root node is leaf

static void printTreeList(**list**<tree*> trees) - Print list of trees

int getValue() - Get the value of the root node

void setValue(**int** val) - Set the value of the root node

int getWeight() - Get the weight of the root node

void setWeight(**int** val) - Set the weight of the root node

node* left() - Get pointer to the left child

node* right() - Get pointer to the right child

Project made with Code::Blocks
by Iliya Zhechev