**SSP Term 7**

**Algorithms**

**Files Compression and Decompression**

By

Mohamed Hicham Gaballah 4683

Hana Hesham Nazmy 4952

Aya Ahmed Kandil 4699

**Node Class**

These are the nodes created while building the Hoffman Tree.

Each node has its frequency (how many times is it repeated in the whole file), value (the character/number being read), left pointer to left child and right pointer to right child (initially null).

**Compress Class**

Compress Constructor:

It calls *readFrequencies* to calculate the frequencies, then builds the Hoffman tree, get the codes for each character, writes the tree into the file as a header, and writes the compressed file itself.

ReadFrequencies:

It reads the file and creates a hashmap for each read int and increments its frequency.

BuildTree:

It creates nodes for each entry in the hashmap created at *ReadFrequencies* then adds them to a Heap, then build the Hoffman Tree by popping 2 minimum Nodes, create a new third node, assign left and right children, and add it to the Heap again with a frequency equal to the sum of the first 2 nodes.

Assign:

It concatenates the value of the code to be used in compression as it assigns a zero to the left node and a one to the right node.

GetCodes:

It creates a hashmap to store the codes of each character read.

WriteTree:

It writes in the compressed file the code as zeros and ones.

Writefile:

It reads the input file, converts every byte to its code, calculates the padding of the last byte, and writes the code for each byte to the compressed file with the padding value.

**Decompress Class**

BuildTree:

It reads the compressed file header which is zeros and ones, creates a node for each read character, and builds a tree.

ReadTree:

It reads the built tree and gets the root.

Inorder:

It prints the codes of the tree in an inorder traversal.

Output:

It reads byte by byte, converts them to binary strings and removes the padding of the last byte. Then, it calls *decode* for every string read of size 8, moves with an index for decoding all 0&1 from the read file. Decode is called a last time for the last byte after removing padding.

Decode:

It traverses the tree in order to get the value of the node being searched for.

**Notes:**

* We used DataInputStream and DataOutputStream in order to be able to read, write, compress and decompress binary files, not only the text files.