CS 5329 – Programming Project – Group 2

Submitted by
Divya Viswanath (A04859120)
Divya Prakash Sheela (A04843101)

Guided by
Dr.Vangelis Metsis
Assistant Professor
Department of Computer Science
Texas State University

1. BST and Heap: Huffman coding and decoding

Project Description

Compress a given text file using Huffman Coding and decode it back to get the original text file.

Inputs

- Input text file which needs to be compressed.

Outputs

- Prints the generated Huffman code for each character
- Compressed file encoded using the generated Huffman code
- Decoded text file
- Outputs the following statistics :-
    - Size of the file before Compression
    - Size of the file after Compression
    - Compression Achieved in terms of Percentage

Program Analysis

General Flow of Program

1. Reads the input file and copy its content to a String
2. Extracts each character from the string store it into a character array.
3. Finds the frequency of each character and store them in an integer array at the corresponding indices as the character array
4. Insert each character into a priority queue according to the frequency
5. Take the lowest two frequencies and find their sum
6. Insert into priority queue a new node with frequency as the calculated sum and remove the lowest two nodes.
7. Repeat the Steps 5 and 6 until the heap is empty
8. Build the path of Huffman code by assigning 0 to left and 1 to right and save the created code into an array at corresponding indices as the character array
9. Insert each character and their Huffman codes into a BST
10. Search the BST for each character in the input text and write the codes into an output file
11. Print out the Huffman codes for each character
12. A dot file will be generated with the formed BST
13. Create a bin file for the Huffman code
14. Decode the file containing the codes using the Huffman code by tracing back the path.
15. Convert the codes back to the corresponding characters and write them into an output file.

Data Structures and Algorithms used in the Program

- The program uses the following Algorithms –
    - MinHeapify
    - Huffman coding algorithm
    - Binary Search tree algorithm
- Arrays, Linked lists are the main data structures used

Analysis of Program

- It took 50.064405 ms to encode a file of size 17984 bits
- A compression rate of 43.44 was obtained.
- It took 45.736861 ms to decode the file
- The complexity of the algorithm will be O(n log n)

Steps to execute the file

- Download Java Development Kit8 from the link provided here https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
- Download the Netbeans IDE from the link provided here https://netbeans.org/downloads/
- Open Netbeans and select file in the menu available at left top corner
- Select open project from menu list and select the file provided
- Select the project and double click to open and build the project by selecting play button available in menu above.
- To see the BST graph go to command line, install GraphViz and give the command

  dot -Tpng BSTtree.dot > output.png

2. Communication Network: Minimum Cost Spanning Tree

Project Description

Generate MST for a given graph. Find conditions under which the change in the Edge's weight affects/does not affect the MST generated above.

Inputs

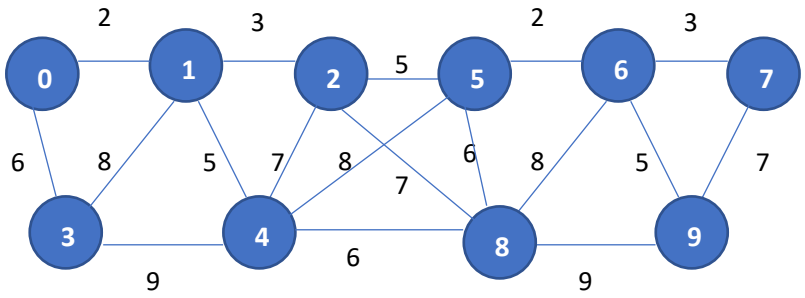
- Input file that contains full graph given in the form of adjacency matrix.
- User's input on the sub-graph selection
  - The row and column taken from the above graph)
- User's input on the new cost of the edge.
  - The vertices of the edge
  - Enter the new cost

Outputs

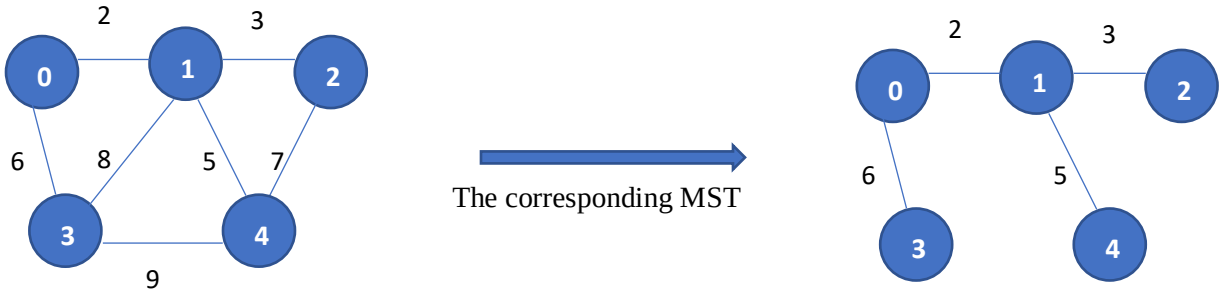
- Produces MCST for the subgraph
- Outputs if the changed weight affects existing MST or not and also outputs the newer MCT if affected.

Graphical Representation of the input given (for immediate Reference)

Input graph given in the attached source code



Eg - Sub graph (If Start Vertex is 0 & End Vertex is 4)



Program Analysis

General flow of the program

- The programs uses the input file to process and store the graph information.
- It prompts the user to choose vertices to form subgraph
- It then proceeds to show the MST for this subgraph.
- Next, it prompts the user which a choice if he/she wishes to change the weight of any edge of the main graph. If so, the user must select the edge he/she wishes to change.
- With the help of the input, it decides whether this change affects the MST or not. If it affects the MST, it recreates MST with this change & displays to the user. There are 4 possible cases:-
    - If edge belong to MST
        - If the old_weight is > new_weight
            - Since the edge is a part of MST and its weight is decreased, this decrease will not affect the MST in any way.
        - If old_weight < new_weight
            - Since the edge is part of MST and its weight is increased, there is a possibility that there would be another MST with a lower weight(which will not include this edge). So, MST will be affected.
    - If edge does not belong to MST
        - If the old_weight > new_weight
            - Since the edge is not a part of MST and its weight is decreased, this decrease may incur another MST that would perhaps include this changed edge. Hence a new MST is possible.
        - If old_weight < new_weight
            - Since the edge is not part of MST and its weight is increased, this increase will not affect the existing MST.

Data Structure and Algorithms used in the Program:-

- The program uses Prims Algorithm to generate MST.
- Arrays are mainly used in this Program

Analysis of the Program.

The complexity of the program would remain $O(V^2)$ as the algorithm used is Prim's Algorithm with the Data Structure Arrays.

Steps to execute the file

- Download Java Development Kit8 from the link provided here https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
- Download the Netbeans IDE from the link provided here https://netbeans.org/downloads/
- Open Netbeans and select file in the menu available at left top corner
- Select open project from menu list and select the file provided
- Select the project and double click to open and build the project by selecting play button available in menu above.
- Provide the inputs required from user in the output panel available below.