# CSE 222 HOMEWORK #6

**Emre BAYRAM 141044019**

**TABLE OF CONTENTS**

1. [REQUIREMENTS](#_TOC_250027) 2
   1. [OVERALL DESCRIPTION](#_TOC_250026) 2
   2. [REQUIREMENT DEFINITIONS](#_TOC_250025) 2
2. [Analysis and Solutions Approach](#_TOC_250024) 3
3. [Class Diagrams](#_TOC_250024) 4
4. [Tests](#_TOC_250024) 5
5. Result…………………………………………………………………………………………………….8

### 

### **Requirements**

## Overall Description

### **Part1**

### Implementing Encode Method it require 2 parameters which these are message which is going to encode, huffmanTree which is used for create a code using this parameter.

### 

### **Part2**

### Implementing iterator class for BinarySearchTree class which is given in source files. This iterator class traverses ascending order.

### 

### **Part3**

### Implementing Priority Queue class using Our Queue interface which same as Java's queue interface. It stores any type of data. When creating order uses compartor given when constructing if it is not given it uses one of the elements compareTo Method and orders bigger to less. Lastly it uses arrayList data structure . After implementing that class created Test class written to create Table of benchmark of work time.

## Requirement Definitions

* + 1. Huffman data when creating Huffman Binary Tree
    2. An entries for Priority Queue

**2. Analysis and Solutions Approach**

**PART 1**

Huffman tree is a Binary Tree so generally we use recursion in Binary Tree. It simple to implement. So that I used Recursion thinking . For every character in message traversed tree and one by one created encoded the message.

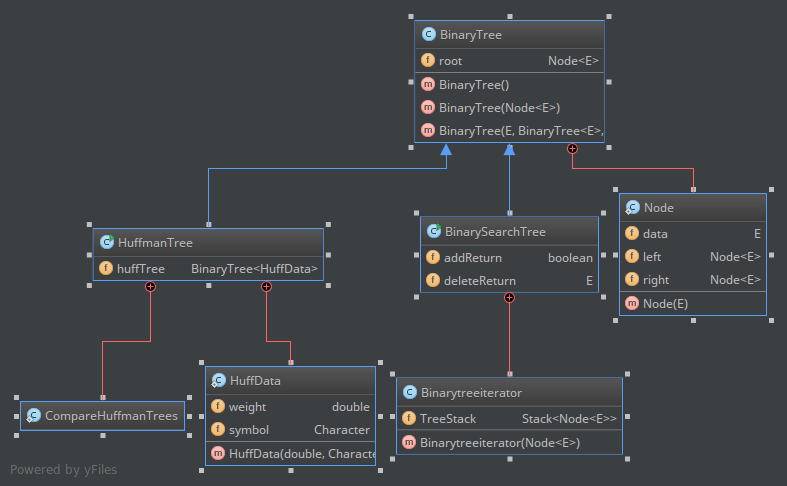
**PART 2**

Best way to traverse is using Stack data structure with iterator. So ı used Stack for store the Tree. Again best way to reach elements is recursion technique When Building a Stack. So I used recursion.

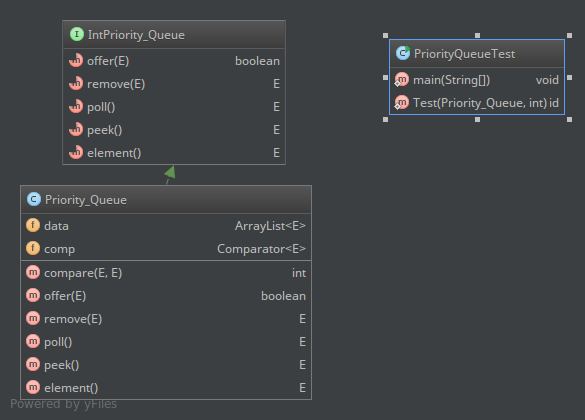
**PART 3**

Priority Queue is like List. There is only one difference between list. This is just like ordered List actually elements has Priority in Queue. So I used ArrayList Data structure when implementing.

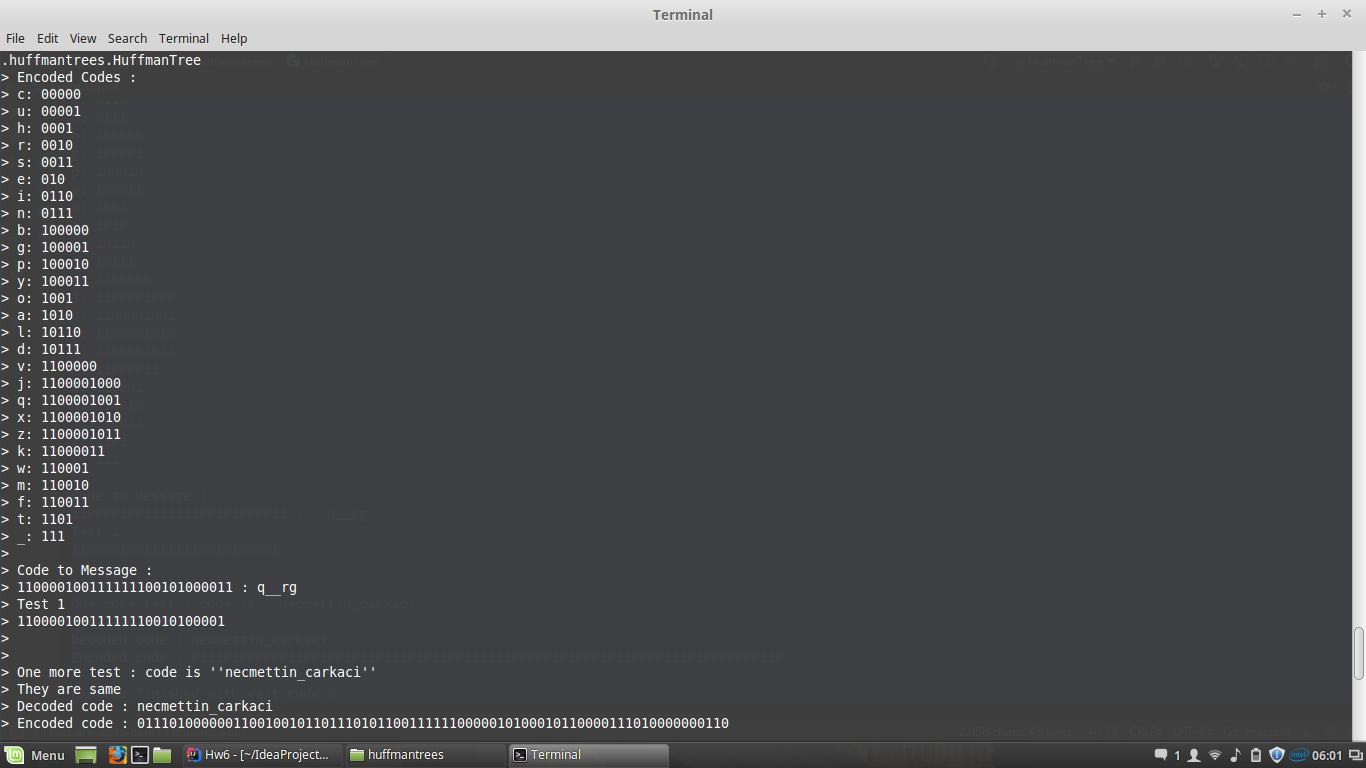
**3. Class Diagrams**



**Figure 1 Binary Tree and Huffman Tree Class Diagrams**

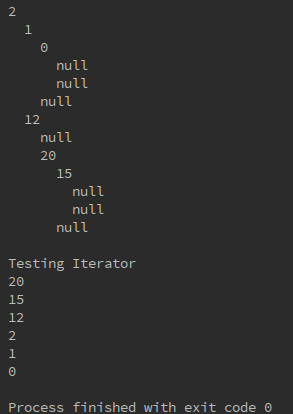


**Figure 2 Priority Queue Class Diagram**

**4. Tests**

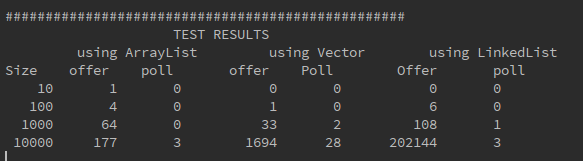
Part 1 Test Results.

Everything is ok.



Part 2 Test Result

Everything is ok.



Part 3 Test Results

Everything is ok.

**5.Result**

As a result I challenged part1. And I couldn't implement the Priority Queue with Binary Tree Because of my other Homework. This homework improve my recursion ability. Linked List class in The test squence take too much time with 100000 test size so I cut it. Vector and arrayList nearly same but linkedList takes too much time.