**From the following list of Projects, choose one of them as your project #2:**

* Morse code (check the table below) is a common code that is used to encode messages consisting of letters and digits. Each letter consists of a series of dots and dashes; for example, the code for the letter a is •– and the code for the letter b is –•••. Store each letter of the alphabet in a node of a binary tree of depth 4. The root node is at depth 0 and stores no letter. The left node at depth 1 stores the letter e (code is •) and the right node stores the letter t (code is –). The four nodes at depth 2 store the letters with codes (••, •–, –•, ––). To build the tree (check the figure below), read a letter followed by its code. The letters should be ordered by tree depth. To find the position for a letter in the tree, scan the code and branch left for a dot and branch right for a dash. Encode a message by replacing each letter by its code symbol. Then decode the message using the Morse code tree. Make sure you use a delimiter symbol between coded letters. [1 or 2 or 3 students].





* Write an application which uses the Huffman\_Tree data structure. Your application will need to read the text file “Sakiro.txt” and build a frequency table for the characters occurring in that file. Convert all the upper-case letters to lower-case when building the frequency table. Once that table is built, create a Huffman code tree and then use it to create the code table. Use a vector of pairs to create the frequency table <char, int> and a vector of pairs to create the code table <char, string>. Make sure to have methods to print those tables in you project. Test your project by converting the following sentence to a binary code and then convert the binary code back to a string:

“bloodborne is going to remain my favorite game for a while longer. bloodborne was my introduction to from software games and the story, atmosphere, and design are great”

**Project Guidelines**

You must follow these guidelines for all the projects that you work on in the data structures course.

**1. Submissions:** You need to submit the following 2 items:

* Source code: Compress your project folder. The folder must consist of all your source code files. I will use this folder to run your application.
* Project Report: This report contains assumptions, a UML diagram (Use the UML power point file for more info about UML), Big-O of the significant functions, and references. This report should be roughly between 2 pages (excluding the front-page).

**2. Source code**

* You can use C++, Java, or C# to implement your project.
* You MUST follow OOP style.
* You need to submit all the files that are needed to run the project. In C++, for instance, you will need to submit header files (.h) and cpp files.
* Comment your code and choose meaningful variable and function names.
* Follow a coding style consistently.
* Don’t include the source code in your printed report. I will just look at it electronically.

**3. Grading Policy**

Factor Percentage

Whether the source code can compile, run, and do what is required. 60%

Class diagram 10%

Quality of algorithms: 30%

* Good performance of the functions.
* Well-documented source code.
* Meaningful variable names.
* Object-oriented source code.