

## CS211 ALGORITHMS & DATA STRUCTURES II

### LAB 1

Dr. Phil Maguire

#### **BINARY TREES**

#### **Pen and Paper Exercise**

Add the following nodes into a binary tree and draw it on paper:

26 58 84 81 94 69 15 29 39 52 33 85

Now delete them in the same order as you added them.

For example, delete 26 first.

Then delete 58.

Then delete 84.... and so on until the tree is empty.

Show how the tree updates following each deletion.

#### **Programming Exercise**

This programming exercise is the first part of creating a Huffman encoding program. You will continue it next week. Write a Java program which takes in a line from the user (using Scanner) and then outputs

- i) The sentence in ASCII
- ii) Each letter in the sentence and its frequency

## **Hints:**

In order to convert to ASCII, go through each letter, cast it to an `int` and then use `Integer.toString()` to convert it to a binary string. Watch out for missing leading zeroes!

```
(int)sentence.charAt(i);
```

```
Integer.toString(someInt);
```

In order to obtain the frequencies for each letter, you could create an array of size 256, and use it to store the frequencies of every ASCII character. Go through the sentence, incrementing the appropriate slot in the array each time. Then print out all the non-zero frequencies in the array.

```
int[] ASCIIarray = new int[256];
```

Your program output should look something like this:

```
Enter your sentence: The cat sat on the mat
```

```
1010100 1101000 1100101 0100000 1100011 1100001 1110100
0100000 1110011 1100001 1110100 0100000 1101111 1101110
0100000 1110100 1101000 1100101 0100000 1101101 1100001
1110100
```

```
' ' appeared 5 times
'T' appeared 1 time
'a' appeared 3 times
'c' appeared 1 time
'e' appeared 2 times
'h' appeared 2 times
'm' appeared 1 time
'n' appeared 1 time
'o' appeared 1 time
's' appeared 1 time
't' appeared 4 times
```