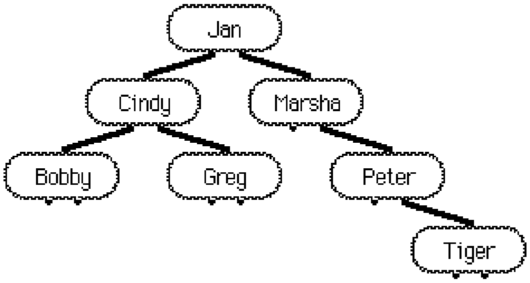
CS106B
Spring 99-00

Handout #31
May 8, 2000

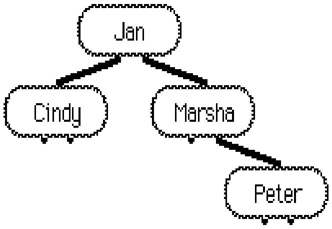
Section Handout #6

Problem 1: Trimming Leaves

Write a function **`TrimLeaves,`** which will take a binary tree and remove all of its leaves. Thus, given the following tree:



The function will return the following tree:



```
void TrimLeaves(nodeT **tree)
```

Problem 2: Run Length Encoding

Write the **`Encode`** and **`Decode`** functions to implement compression via *run-length encoding*. This technique replaces each repeated sequence of identical letters with the count and letter. The **`Encode`** function takes an input FILE * and output FILE * and compresses according to this technique. An interesting effect is that the encoded file is guaranteed to be either the same length as the original file or shorter, which makes it useful for data compression. Here are some examples:

FILE *in	Encoded FILE *out
`"Hello world"`	`"He2lo world"`
`"Bookkeeper"`	`"B2o2k2eper"`
`"Oooga boooga boooga"`	`"O2oga b3oga b3oga"`

The `Decode` operation reverses this compression. To simplify things, you can assume that there will no be digit characters in the input file. (But as a though experiment -- how could you extend your encoding to remove this restriction?)

Problem 3: Huffman Coding

So, to make sure you understand how the algorithm for building an optimal Huffman encoding tree, come up with a valid tree for the following string:

“cs106b is sick"

Problem 4: Thought Questions

a) Having done both run-length encoding and huffman encoding now, when would one method be more advantageous than the other? For what type of files would huffman encoding be good for? What kind of files would it actually "compress" into a larger file?

b) What would happen if you ran the huffman compression on a file you already compressed with the same program? Would it get any smaller? Why or why not?

c) Since certain letter pairs such as "th" and "ch" appear much more commonly in English text pairs such as 'xs" and so on, you could also do huffman encoding where instead of encoding and counting single characters, you instead encoded by consider pairs of letters and their frequencies. Discuss the various advantages and drawbacks of doing this over single characters.