**Homework 6**

Due Date: July 20

8 Points Total (Extra Points Available)

As the end of Chapter 7 shows us, there are many applications/uses of trees. You’re going to develop a solution for one such application.

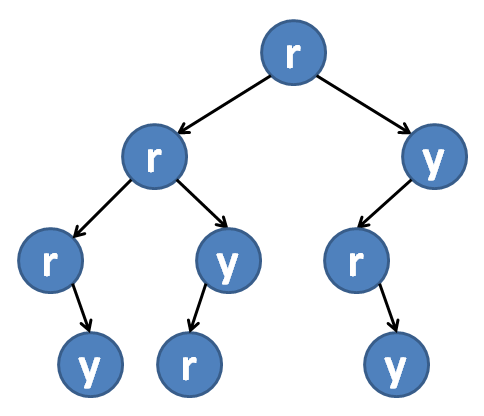
There are analogous problems in other disciplines, like recognizing the source of a text based upon grammatical and word-usage or determining the path of a video game; nevertheless, we’ll look at an example in molecular biology. So organisms have genomes and these genomes are made up of nucleotides. There are 4 (A, T, C & G). In other words, the alphabet of life is just four characters. Okay, so these nucleotides belong to one of two groups – purines (A & G) and pyrimidines (C &T) – which we often abbreviate in a 2 letter code as R and Y respectively. Let’s say that we have the complete genome sequence for some horrible virus strain. We then are sampling people showing symptoms and get fragments (aka words) of the virus which infects these. (Think the movie *Contagion*.) Anyways, our goal is to determine if the patient in fact has the terrible virus or just some boring one like a cold or the flu. We can solve this problem using a tree.

***How??***

1. Take the genome sequence and break it up into all words.

2. Transform these words into a tree.

Let’s see for example, if I have the words rrry, rryr and ryry, I could represent them with this tree:

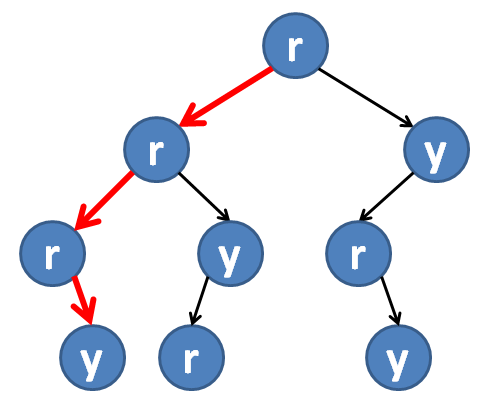
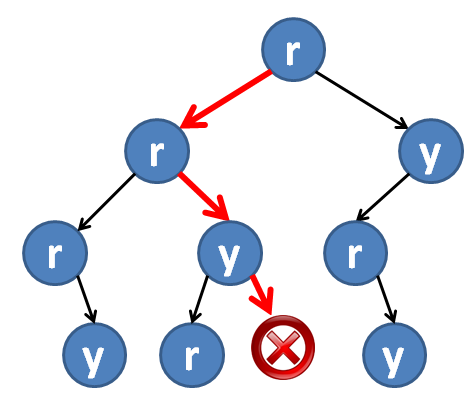


A few things to note. The link between parent and child implies that the letter of the child is seen subsequent to the letter of the parent in the word.

Secondly, notice that r followed by y occurs three times in the tree. A space efficient tree would link back to such occurrences. This is complicated. Let’s just be simple.

3. Take in a file of reads which are of the same length as the words defined in step 1 and represented as a tree in step 2.

Let’s say I have the sequence rrry. Is it in the genome? Is rryy in the genome?

No it isn’t. While we have two of the three edges, there is no evidence (built from genome sequence) of rryy.

Yes it is! Follow the path shown by the red arrows.

To get you started, I’ve created two classes BTreeNode and QTreeNode.

**class BTreeNode**

**{**

**public:**

**char Data;**

**BTreeNode \*Lchild;**

**BTreeNode \*Rchild;**

**};**

**class QTreeNode**

**{**

**public:**

**char Data;**

**QTreeNode \*child1;**

**QTreeNode \*child2;**

**QTreeNode \*child3;**

**QTreeNode \*child4;**

**};**

I’ve also created two functions get\_words(…) and get\_reads(…) which will get the words from the genome file and reads from the file respectively. These functions are written to read in the files of a 4 character alphabet and either convert them to a two character alphabet (in the case that you will be using BTreeNode) or stay as a four character alphabet (in the case that you will be using QTreeNode).

Your task is to develop a class which either uses BTreeNode or QTreeNode that will take the words from the vector created by the get\_words(…) function and create either a binary tree or a quaternary tree using BTreeNode or QTreeNode, respectively. This class will then have functionality to traverse particular paths through the tree, as specified given a read contained within the vector created by get\_reads(…). The file will be generated as specified previously. The number of reads that map (ie, “Yes”) and the total number of reads will be reported to the user via the console.

In addition to your code file(s). Your submission must also include a ½- ¾ page report. In this report, you must clearly state:

* If your functionality is for the 2 character or 4 character alphabet (or both)
* The IDE you used to develop your code
* A screen-shot of your final code compiled (showing no errors) in your IDE
* A screen-shot of the console window showing the results requested
* A 1 paragraph summary of how your code works. (Think of this as some documentation for the user.)
* A list of all the files included in your submission and a brief discussion of what they are.

**Grading:**

Developing an application for the BTreeNode, 2 character alphabet, is simply a modification of the binary tree functionality discussed during class and presented in the textbook and in the notes. This is the most straightforward option for this assignment. As such, it is worth **8 points total**:

2pts Functionality to create the tree correctly.

2pts Functionality to trace through tree for a given read.

2 pts Functionality to write out results to user regarding each read’s presence/absence.

1pt Quality of commenting.

1 pts Report. (This is an all or nothing grade… no partial credit.)

If you develop an application for the QTreeNode, 4 character alphabet, it is more of a challenge. As such it is worth **6 points total**:

3pts Functionality to create the tree correctly.

1.5pts Functionality to trace through tree for a given read.

0.5 pts Functionality to write out results to user regarding each read’s presence/absence.

0.5pt Quality of commenting.

0.5 pts Report. (This is an all or nothing grade… no partial credit.)

If you develop an application which takes a user input to determine if it will work with a 2 character or 4 character alphabet, ie it is developed for BOTH cases (2 classes), then it is worth **14 points total**:

**Additional comments re: grading:**

1. If your code has any missing semicolon/paren/bracket/brace, misspelled user-created function call or variable name, or incorrect for/while/if/else syntax which prohibits it from compiling, **you will receive a zero for the entire assignment**. These are simple syntax issues that you should be able to readily identify and fix.
2. Notice that you will gain a point for the “quality of commenting”. This assumes commenting. If the only comments which appear in your file are the ones I have included to start you out, **you will receive a zero for the entire assignment**. This is poor programming.
3. If you code compiles but does not work as it should, as discussed here, **you will receive at most 4 points**, regardless of if your implementation was for binary, quaternary or both.

**PLEASE TAKE ALL OF THE FILES TO BE INCLUDED IN YOUR SUBMISSION AND EITHER COMPRESS THEM AS A ZIP OR RAR FILE SUCH THAT YOUR SUBMISSION IS JUST THE ONE COMPRESSED FOLDER, THE NAME OF WHICH IS YOUR FIRST INITIAL FOLLOWED BY YOUR LAST NAME (ie, my submission would be cputonti.zip or msaenz.rar). This greatly expedites grading as Sakai changes the names of files. The goal is to have this assignment graded and back to you before the Withdraw date.**

**Helpful hints.** Read this assignment and the code provided several times. Map out a plan on paper to make certain that you understand both the problem and the approach you will take before you even start coding!!! Ask questions EARLY.