CS106B
Spring 2013

Handout #10
April 15, 2013

Section Handout 2

This week's section handout is all about collections. There are two problems here – one that probes the mysteries of the English language, and one that probes the mysteries of life itself.

Problem One: Xzibit Words

Some words contain other words as substrings. For example, the word "pirates" contains a huge number of words as substrings:

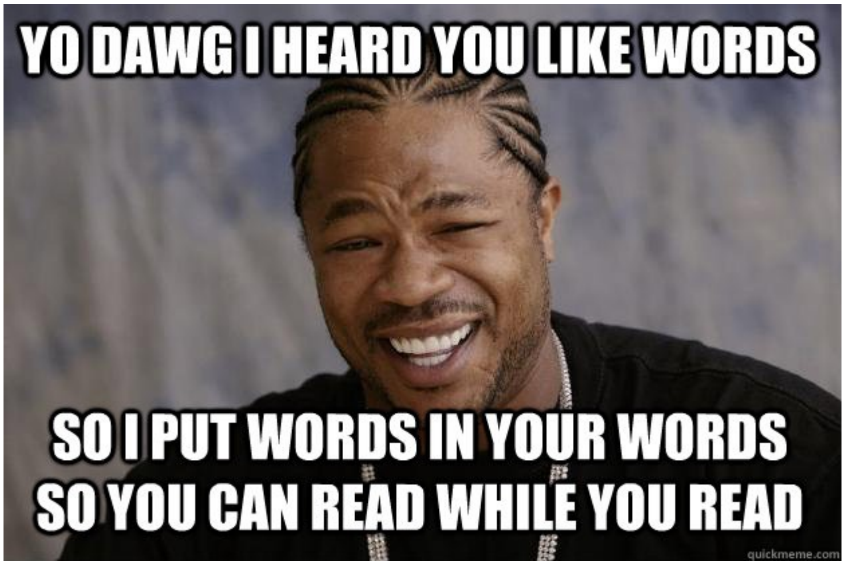
a
at
ate
ates
es
I
irate
pi
pirate
pirates
rat
rate
rates

Note that "pirates" is a substring of itself. The word "pat" is not considered a substring of "pirates," since even though all the letters of "pat" are present in "pirate" in the right order, they aren't adjacent to one another.

Let's call a word an "Xzibit word" if it contains a large number of words as substrings. Write a function

```
string mostXzibitWord(Lexicon& words);
```

that accepts as input a **Lexicon** of all words in English, then returns the word with the largest number of words contained as substrings.


YO DAWG I HEARD YOU LIKE WORDS
SO I PUT WORDS IN YOUR WORDS
SO YOU CAN READ WHILE YOU READ
quickmeme.com

Problem Two: RNA Protein Codes

RNA (ribonucleic acid) is a chemical that can encode genetic information. Plant and animal cells use RNA for a variety of cell functions, while viruses often use RNA as their primary genetic storage.

Each strand of RNA consists of a series of nucleotides – adenine (A), guanine (G), uracil (U), and cytosine (C) – and a strand of RNA can be thought of as a string over those four letters. Because of this, computational geneticists often treat DNA and RNA as strings and run algorithms on those strings to deduce their properties.

RNA is used by a cell to encode **proteins**, biological molecules essential to cell function. Each protein consists of a series of **amino acids** that, strung together, serve some complex purpose. The actual letters in an RNA strand spell out what amino acids need to be combined together to produce the overall protein. So how exactly are these amino acids represented? In RNA, letters are grouped into clusters of three letters called **codons**. Each codon encodes a specific choice of amino acid. When decoding RNA, the cell reads these codons in sequence and assembles the protein one amino acid at a time. For example, the RNA strand

GGGAUGAAUAUCUCGGCG

would be treated at this sequence of three-letter codons:

GGG   AUG   AAU   AUC   UCG   GCG

The cell would then use the codons to determine what amino acids to string together into a protein. In this case, these codons represent the following sequence of amino acids:

GGG	AUG	AAU	AUC	UCG	GCG
Glycine	Methionine	Asparagine	Isoleucine	Serine	Alanine

So the generated protein would have amino acids ordered as glycine, then methionine, then aspargine, then isoleucine, then serine, and finally alanine.

An important detail is that each strand of RNA does not encode just one protein; typically, a single strand of RNA encodes many different proteins. How, then, does the cell know where each protein starts and stops? There is an ingenious system set up for just this purpose. In an RNA strand, the codon AUG has two meanings – it can code for methionine (as shown above), but it also acts as a special "start of protein" marker. As a cell scans across an RNA strand, if it encounters the codon AUG, it begins assembling a protein starting at that location. It then continues to assemble the protein one amino acid at a time by decoding codons in sequence. The cell stops assembling base pairs when it encounters one of three "stop" codons (UAA, UAG, or UGA) that act as an "end-of-protein" marker. The cell can then keep scanning across the RNA until it finds another AUG start codon, at which point the process repeats. For example, consider this RNA strand:

GCAUGGAUUAAUAUGAGACGACUAAUAGGAUAGUUACAACCCUUAUGUCACCGCCUUGA

This would be decoded as follows:

GC	Skip letters until we find AUG.
AUGGAU**UAA**	Read from AUG until we hit a stop codon (here, UAA)
U	Skip letters until we find AUG.
AUGAGACGACUAAUAGGA**UAG**	Read from AUG until we hit a stop codon (here, UAG)
UUACAACCCUU	Skip letters until we find AUG.
AUGCACCGCCU**UGA**	Read from AUG until we hit a stop codon (here, UGA)

Your job is to write a function

```
Vector<string> findProteins(string& rna, Map<string, string> codons);
```

that accepts as input a string of RNA and returns a `Vector` of all the proteins that would be formed from that RNA. The first parameter represents the actual string of RNA, and the second parameter is a `Map` from three-letter codons to the name of the amino acid they represent (or the special string `"stop"` if the codon is a stop codon). For example, running this function on the above RNA strand would return the `Vector` holding the stings

methionine, asparitic acid	(encoded by AUGGAU**UAA**)
methionine, arginine, arginine, leucine, isoleucine, glycine	(encoded by AUGAGACGACUAAUAGGA**UAG**)
methionine, serine, proline, proline	(encoded by AUGUCACCGCCU**UGA**)

You can assume that all the proteins are properly terminated, which means that if you find an AUG codon then there will be a matching stop codon before the end of the protein.