Assignment 4a
SOLUTIONS

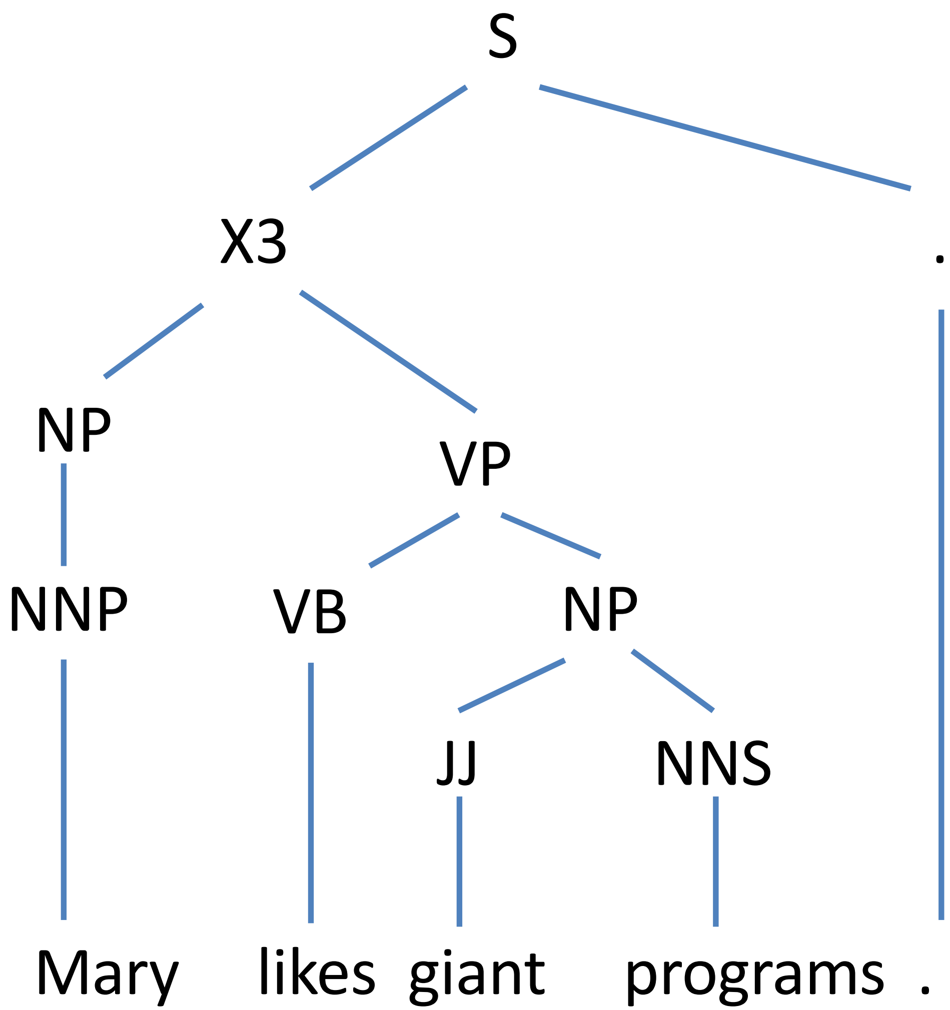
Problem 2a:

Mary	likes	giant	programs	.
NNP: -0.2 NP: -0.35	-	-	X3: -2.05	S: -2.15
	VB: -0.6	-	VP: -1.7	S: -2.3
		JJ: 0.0	NP: -0.8	-
			NNS: 0.0	-
				.: 0.0

With backpointers:

Mary	likes	giant	programs	.
NNP: -0.2 NP: -0.35	-	-	X3: -2.05	S: -2.15
	VB: -0.6	-	VP: -1.7	S: -2.3
		JJ: 0.0	NP: -0.8	-
			NNS: 0.0	-
				.: 0.0

Problem 2b:



Problem 3:

I will give individual feedback as I grade. There are many ways to solve this, so I'm not going to give a particular solution. If you have questions about your particular implementation, come talk to me.

Problem 4:

It can be a bit counterintuitive, however, option b, simply iterating through all the grammar rules tends to be faster in practice for large grammars assuming that you can lookup quickly (i.e., O(1)) whether a constituent exists in an entry in your table.

The numbers can help explain this. For the first sentence, if we have on average 446 constituents in a given entry in the table, if we're considering all possible pairs, then we will on average have to consider 446^2 (~200K) possibilities. Larger entries are particularly problematic, for example, the worst case would be 916**2 (~840K) combinations. On the other hand, there are only 51K binary rules. Therefore, on average, option a will generally be at least 4 times slower, but often much worse since it will be very slow on the larger entries.

If you consider the worst test sentence, this is even worse, with on average 1.2M combinations and almost 6M in the worst case.