CSCE-638, Programming Assignment #3 CKY
Due: Sunday, October 28th, 2018 by 11:00pm

For this assignment, you should write a program that implements the classic CKY algorithm for syntactic parsing. No starter code is provided this time since we expect you can handle file IO etc by now. As we discussed in class, the CKY algorithm is a dynamic algorithm to efficiently find the most likely syntactic parse tree for a sentence. As input, your program will need two files: (1) a file of grammar rules with probabilities, and (2) a file of sentences to process. Your program should read the names of the 2 input files from the command line. The command-line arguments should be in the following order:

```
CKY <grammar_file> <sentences_file>

CKY grammar_rules.txt sents.txt
```

1. The grammar_rule.txt File

This file will contain grammar rules with probabilities in the Chomsky normal form. The rules are exactly the same as the ones we have seen in the in-class example.

2. The Sentences File

This file will contain a set of sentences that you should run your program on. Each sentence will be on a separate line. There will not be any punctuation marks in this file. A sample file might look like this:

> This is a sentence
> And here is another sentence

OUTPUT FORMATTING

The output produced by your program should consist of 2 items for <u>each</u> sentence that is processed:

1. The sentence being processed.
2. All final and intermediate probabilities generated during the parsing process, as shown in the boxes on slide 32 for the example sentence. Note that probabilities in the boxes should be printed in the order of outside-to-inside, left-to-right.

IMPORTANT: When you print this information, please format it <u>exactly</u> like the example in Figure 1 and Figure 2 (cont) on the next page! Since you'll be printing a lot of numbers, this is crucial to ensure that we know exactly what your numbers and results correspond to.

GRADING CRITERIA

Your program will be graded based on both given dev cases (50%) and new test cases (50%)! **So please test your program thoroughly to evaluate the generality and correctness of your code!** Even if your program works perfectly on the examples that we give you, that does not guarantee that it will work perfectly on additional test cases.

ELECTRONIC SUBMISSION INSTRUCTIONS
(a.k.a. "What to turn in and how to tun in")

You need to submit 2 things:

1. The source code files for your program. Be sure to include all files that we will need to compile and run your program!
2. A report file that includes the following information:
   - how to compile and run your code
   - results and analysis
   - any known bugs, problems, or limitations of your program

   REMINDER: your program **must** compile and run. We will not grade programs that cannot run.

How to turn in:

1. please include everything in a single .tar.gz (.tgz) package and submit it via ecampus.

PROCESSING SENTENCE: fish people fish tanks

SPAN: fish
P(N) = 0.2
P(V) = 0.6
P(NP) = 0.14 (BackPointer = N)
P(VP) = 0.06 (BackPointer = V)
P(S) = 0.006 (BackPointer = VP)

SPAN: people
P(N people) = 0.5
P(V people) = 0.1
P(NP) = 0.35 (BackPointer = N)
P(VP) = 0.01 (BackPointer = V)
P(S) = 0.001 (BackPointer = VP)

SPAN: fish
P(N fish) = 0.2
P(V fish) = 0.6
P(NP) = 0.14 (BackPointer = N)
P(VP) = 0.06 (BackPointer = V)
P(S) = 0.006 (BackPointer = VP)

SPAN: tanks
P(N tanks) = 0.2
P(V tanks) = 0.3
P(NP) = 0.14 (BackPointer = N)
P(VP) = 0.03 (BackPointer = V)
P(S) = 0.003 (BackPointer = VP)

SPAN: fish people
P(NP) = 0.0049 (BackPointer = (1, NP, NP))
P(VP) = 0.105 (BackPointer = (1, V, NP))
P(S) = 0.0105 (BackPointer = VP)

SPAN: people fish
P(NP) = 0.0049 (BackPointer = (2, NP, NP))
P(VP) = 0.007 (BackPointer = (2, V, NP))
P(S) = 0.0189 (BackPointer = (2, NP, VP))

SPAN: fish tanks
P(NP) = 0.00196 (BackPointer = (3, NP, NP))
P(VP) = 0.042 (BackPointer = (3, V, NP))
P(S) = 0.0042 (BackPointer = VP)

Figure 1: Sample Output for CKY Program

SPAN: fish people fish
P(NP) = 0.0000686 (BackPointer = (1, NP, NP))
P(VP) = 0.00147 (BackPointer = (1, V, NP))
P(S) = 0.000882 (BackPointer = (1, NP, VP))

SPAN: people fish tanks
P(NP) = 0.0000686 (BackPointer = (3, NP, NP))
P(VP) = 0.000098 (BackPointer = (2, V, NP))
P(S) = 0.01323 (BackPointer = (2, NP, VP))

SPAN: fish people fish tanks
P(NP) = 0.0000009604 (BackPointer = (1, NP, NP))
P(VP) = 0.00002058 (BackPointer = (1, V, NP))
P(S) = 0.00018522 (BackPointer = (2, NP, VP))

Figure 2: Sample Output for CKY Program - cont.