CS 1.2: Intro to Data Structures & Algorithms

Histogram & Markov Chain Worksheet

Name: ________________

Text: *"I like dogs and you like dogs. I like cats but you hate cats."* (ignore all punctuation)

Histograms

Q1: How many distinct word *types* are present in this input text? How many total word *tokens*?

Distinct word types: 8   Total word tokens: 14

Q2: What data structure would be appropriate to store a histogram counting *word frequency*? Why did you choose this data structure? In other words, what makes this data structure ideal?

A dict is a good data structure becuase it is easy to access keys (words) and values (frequency)

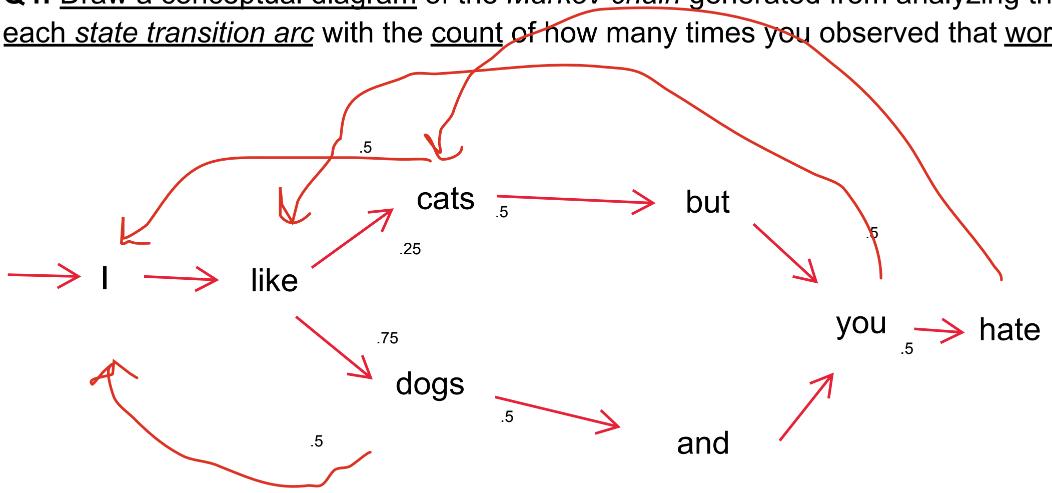
Q3: Write the data structure you would create to store this histogram counting *word frequency* (as it would look if you printed it out with Python).

```
for word in word_list:
        self[word] = self.get(word, 0) + 1
```

key: word, value: frequency

Markov Chains

Q4: Draw a conceptual diagram of the *Markov chain* generated from analyzing the text above. Label each *state transition arc* with the count of how many times you observed that word pair.



Q5: Write the data structure you would create to store the word transitions out of the state that represents the word *"like"* in this Markov chain (as it would look if you printed it out with Python).

I think the data structure would be some sort of tree that we navigate starting at the chosen word ("like") and move by the probabilty coin flip until we create a string of words (the sentence) so like: [.25, cats], [.75 dogs]{

Q6: Write a *new sentence* that can be *generated* by doing a *random walk* on this Markov chain.

I like dogs and you like cats