function: next_seq

1. Inputs: a group of ordered elements, new element
2. Output: the group of elements with the first element omitted and the new element added to the end
3. The input group of elements is converted into a mutable type.
4. The first element from the group of ordered elements is removed.
5. The new element is added to the end of the group of ordered elements.
6. The group of elements is converted back into a non-mutable type.
7. The group of elements is returned.

function: disjoint2

1. Inputs: input set 1, input set 2
2. Output: boolean indicating whether the sets are disjoint or not
3. For every element in the first input set, checks to see if that element is in the second input set.
4. If any element is in both sets, False is returned.
5. If after looping through, False was never returned, then True is returned.

function: random_choice_weighted

1. Inputs: word to probability map, random probability generator
2. Output: word
3. A random probability is generated.
4. Every key mapped to every value in the input mapping is a pair.
5. For every key and every value in pairs, if the probability generated is less than the value, then the key (word) is returned.
6. Otherwise, the probability is reset to probability minus value and the process re-runs.

function: generate_text

1. Inputs: chain of words, number of words, starting function, random function
2. Output: random part of chain of words
3. The starting function selects a portion of the chain of words and it is stored.
4. An empty list (general term) for words to be added to is created.
5. For every word in the portion of chain of words, the word is added to the list.
6. For every other word not in the portion, if the word is succeeded by another, then the random function is used to randomly select the next word.
    a. This word is added to the list.
7. The next_seq helper function is called to get rid of the current word and look at the next word.
8. If the word doesn't have a successor, then the loop stops.
9. An empty string joined to the list of words is returned.

function: neural_net

1. Inputs: input set, output set, # of hidden nodes
2. Output: directed graph
3. There are conditions that must be met before the function can run:
    a. If the number of hidden nodes is < 0, None is returned.
    b. If either set is empty, None is returned.
    c. If the sets are not disjoint (helper function is used), None is returned.
4. An empty map is created to connect input nodes to hidden nodes and hidden nodes to output nodes.
5. Because output nodes are not connected to anything, an empty set is created.
6. For every element in input node set, the input node is mapped to a hidden node.
7. For every element in the hidden node set, the hidden node is mapped to an output node.
8. For every element in the output node set, the output node is mapped to the empty set.
9. The directed map is returned.