Data definitions and headers:

```
IntList : TypeAlias = Optional["IntNode"]
WordLinesList : TypeAlias = Optional["WordLineNode"]


@dataclass
class LLIterator:
   list : IntList | WordLinesList
   def __iter__(self) -> Self:
       return self
   def __next__(self) -> "int | WordLines":
       if (self.list is None):
           raise StopIteration
       to_return = self.list.value
       self.list = self.list.next
       return to_return


@dataclass(frozen=True)
class IntNode:
   value: int
   next: IntList
   def __iter__(self) -> LLIterator:
       return LLIterator(self)


@dataclass
class WordLines:
   word: str
   line_numbers: IntList

@dataclass(frozen=True)
class WordLineNode:
   value : WordLines
   next: WordLinesList
   def __iter__(self) -> LLIterator:
       return LLIterator(self)

@dataclass
class HashTable:
   table : List[WordLinesList]
   count : int
```

```
# Returns a hash of string
def hash_fn(v:str) -> int:

# Makes an empty hashtable
def make_hash(size:int) -> HashTable:

# Return the number of elements in the given hash table
def hash_count(ht:HashTable) -> int:

# Return the size of the given hash table
```

```
def hash_size(ht:HashTable) -> int:

# Does the hash table contain a mapping for the given word?
def has_key(ht:HashTable, key:str) -> bool:

# What line numbers is the given key mapped to in the given hash table?
def lookup(ht:HashTable, key:str) -> List[int]:

# Finds a word line with the given key in a word line list
def find_word_line(word_line_list: WordLinesList, word:str) -> Optional[WordLines]:

# Adds the value to the end of the list if it isn't already in the list
def append_list(list : IntList | WordLinesList, value : int | WordLines) -> IntList |
WordLinesList:

# Adds a word-line mapping to an array
def add_word_lines(array: List[WordLinesList], value :  WordLines) -> None:

# Add a mapping from the given word to the given line number in
# the given hash table
def add(ht: HashTable, word: str, line_number: int) -> None:

# What are the words that have mappings in this hash table?
def hash_keys(ht: HashTable) -> List[str]:

# given a list of stop words and a list of strings representing lines of
# a text, return a hash table
def make_concordance(stop_words: HashTable, text: List[str]) -> HashTable:

# given an input file , a stop-words file, and an output file, overwrite the output file with
# a sorted concordance of the input file.
def full_concordance(in_file: str, stop_words_file: str, out_file: str) -> None:
```

Description of running the program on a large file:

I ran the program on a transcription of Lord of the Rings by J.R.R Tolkien, which is around 3.2MBs and contains 48722 lines. It took a while to run, likely because appending on an immutable linked list has a time complexity of O(n) which builds up as the number of occurrences of a word increases. Additionally I don't think I chose enough stop words as the word 'for' occurred on roughly 4107 lines, and there are likely other words which occur even more frequently. Overall the output file contained 14178 lines, and was around 2.6MBs in size which seems to imply that the average line length is much higher in the output file, likely due to many words with hundreds or thousands of occurrences.
I verified the following lines:

```
welled 2267 27038 32179 43878 44918 45517 45967
wellforged 20650
welling 12741 15563 35732 35914
wellinghall 26448 30215
wellnigh 38041
wells 10311 20210 20262 20449 21982 26193 27104 43594 44352
welter 43293
wen 16979 21931
```

I found that all words occurred in the file however I am off by 1 as my line numbers start at line 0 whereas my IDE starts the line numbers at 1. Since I did not see whether to number lines from 1 or 0 in the assignment description I will ask Professor Philbrick, and potentially resubmit this assignment.

An animal that does not occur anywhere in my input text

I was bored so I decided to check against this list I found on the internet. Ok just kidding I totally just nerd-sniped myself and now I've been inspired to make a little app that shows what animals appear in what texts. That being said, there are no Aardvark in LOTR :(
If I ever end up finishing it, it can probably be found here:
https://github.com/roguefirework/Animals-D