CS532 Homework 7

Archana Machireddy

Question 1

class lexicon:

def \_\_init\_\_(self):

file1 = open('hw7.txt','r')

words = file1.readlines()

self.word\_array = [s.strip('\n') for s in words]

def get\_index(self, input\_string, first, last):

word\_len = len(input\_string)

if last < first:

return last

if word\_len == 0:

return -1

mid = (first + last) // 2

if self.word\_array[mid][0:word\_len] < input\_string:

return self.get\_index(input\_string, mid + 1, last)

elif self.word\_array[mid][0:word\_len] > input\_string:

return self.get\_index(input\_string, first, mid - 1)

else:

while self.word\_array[mid][0:word\_len] == self.word\_array[mid-1][0:word\_len]:

mid = mid - 1

if self.word\_array[mid] == input\_string:

return mid

return mid - 1

def find\_after(self,input\_string):

first = 0

last = len(self.word\_array) - 1

index = self.get\_index(input\_string,first,last)

next\_index = index + 1

if next\_index == last+1:

return None

return next\_index

def next5(self,input\_string):

first = 0

last = len(self.word\_array)

index = self.get\_index(input\_string,first,last)

next5\_words = []

next\_index = index + 1

num = 0

while next\_index < last and num < 5:

next5\_words.append(self.word\_array[next\_index])

next\_index = next\_index + 1

num = num + 1

return next5\_words

def prefix5(self,input\_string):

first = 0

last = len(self.word\_array)

index = self.get\_index(input\_string,first,last)

next5\_prefix\_words = []

next\_index = index + 1

num = 0

word\_len = len(input\_string)

while next\_index < last and num < 5:

if self.word\_array[next\_index][0:word\_len] == input\_string:

next5\_prefix\_words.append(self.word\_array[next\_index])

next\_index = next\_index + 1

num = num + 1

return next5\_prefix\_words

Question 2

def retrieve\_all():

l = lexicon()

all\_words = []

num = 5

a = ''

while num >= 5 :

words\_5 = l.next5(a)

all\_words.extend(words\_5)

num = len(words\_5)

if num >= 5:

a = words\_5[4]

print(all\_words)

assert all\_words == l.word\_array

Question 3

def add\_child(self,other):

other.parent = self

if self.child is None:

self.child = other

else:

ch = self.child

if other.value <= ch.value:

other.sibling = ch

self.child = other

else:

while ch.sibling != None and other.value > ch.sibling.value:

ch = ch.sibling

new = ch.sibling

ch.sibling = other

other.sibling = new

def find(self,string):

while len(string) > 0:

a = len(self.value)

if self.value == string[0:a]:

if self.value == string:

node = self

break

self = self.child

string = string[a:]

else:

self = self.sibling

return node