Kevin Ziemba
001324999
ziembak

CoE SI4 : Lab 1

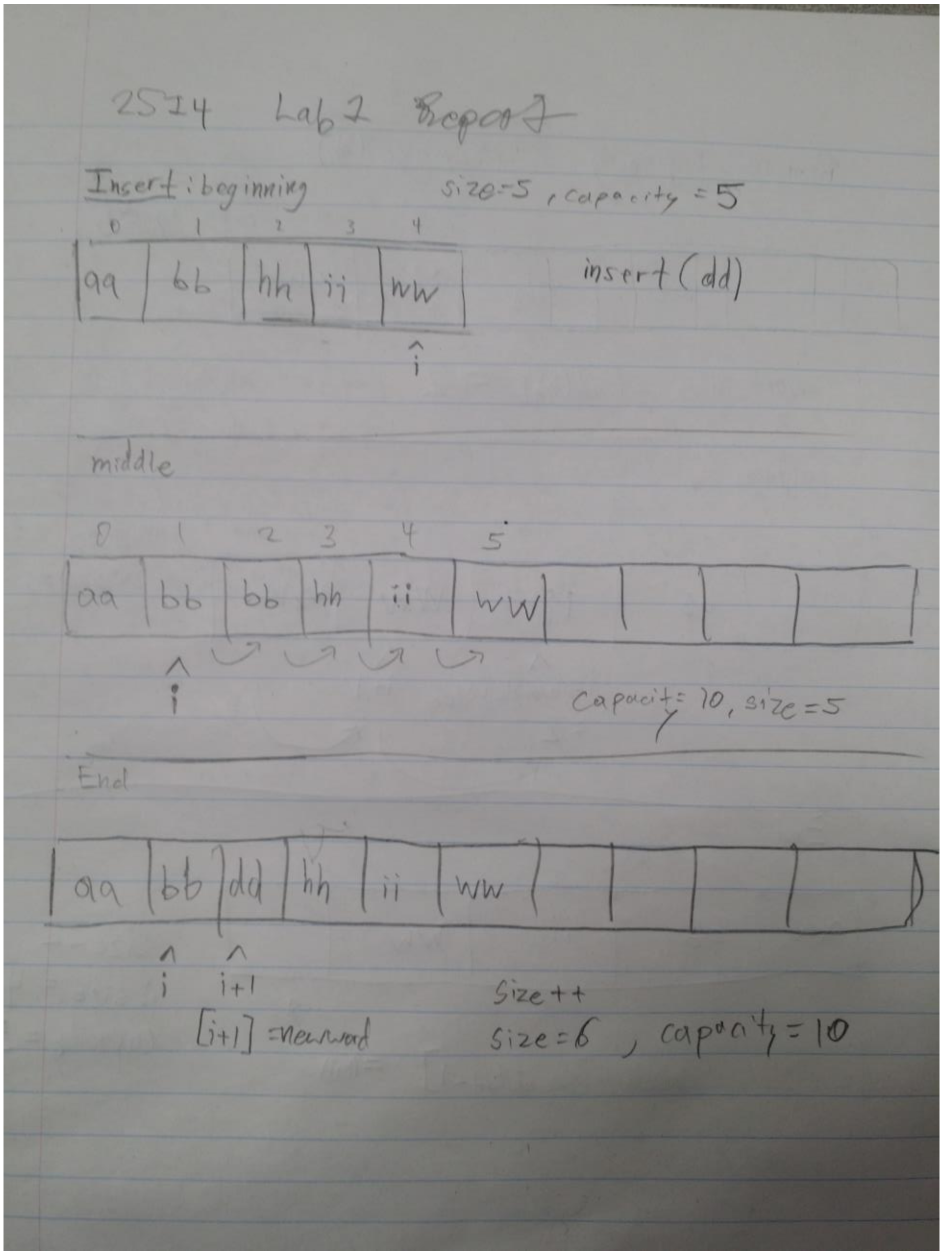
Date Started: Jan 15,2016 Half Finished: Jan 18,2016 Completed Date: Jan 19,2016

Code Descriptions

Method **Insert()** will traverse the WordList from the last element to the first . It will locate the first word in WordList which newword is higher ranked in lexicographical order. Newword is to be inserted after the first word which newword is ranked higher. In order to accommodate the inserted element, all elements which appear after newword are shifted the right, as they are examined. There is temporarily two copies of the same word, but this is overwritten either by a following shift, or by an insertion overwriting one copy. There are handlers for the case where newword is inserted at the end,middle, or start of WordList, as well as for the case no insertion is needed (method **find()** is used). The size of the list is updated if an insertion is made, and if the list is at full capacity, the WordList is copied into a new list with double the capacity.

Method **Remove()** begins by calling method **find()** to obtain the index of the word to be removed. If the word is not in the WordList, nothing is done. Otherwise, the WordList is traversed from the index of the word to be removed, to the end. At each iteration, the current element is overwritten by the next word in WordList. The last element is set to null as the last element as been copied into the second last element, and the size of the list is decremented.

Method **countInRange()** will use binary search to find the *Start* and *End* indices. *Start* is the index of the first element in WordList which the previous element is strictly of lower lexicographical order than character *init*. *End* is the index of the last element in WordList in which the next element is strictly of higher lexicographical order than character *fin*. As such, *Start* will need to examine the previous word to verify its position, and *End* will need to examine the next word throughout each iteration. The difference between End and Start, and then adding 1, is the number of words in range [init,fin]. This method runs in O(log n) as from each iteration traversing through the WordList, the number of future elements that need to be checked is halved.



2514 Lab2 Report

Insert: beginning   size=5, capacity=5

0	1	2	3	4
aa	bb	hh	ii	ww
				^ i

insert(dd)

middle

0	1	2	3	4	5				
aa	bb	bb	hh	ii	ww				
	^ i								

capacit= 10, size=5

End

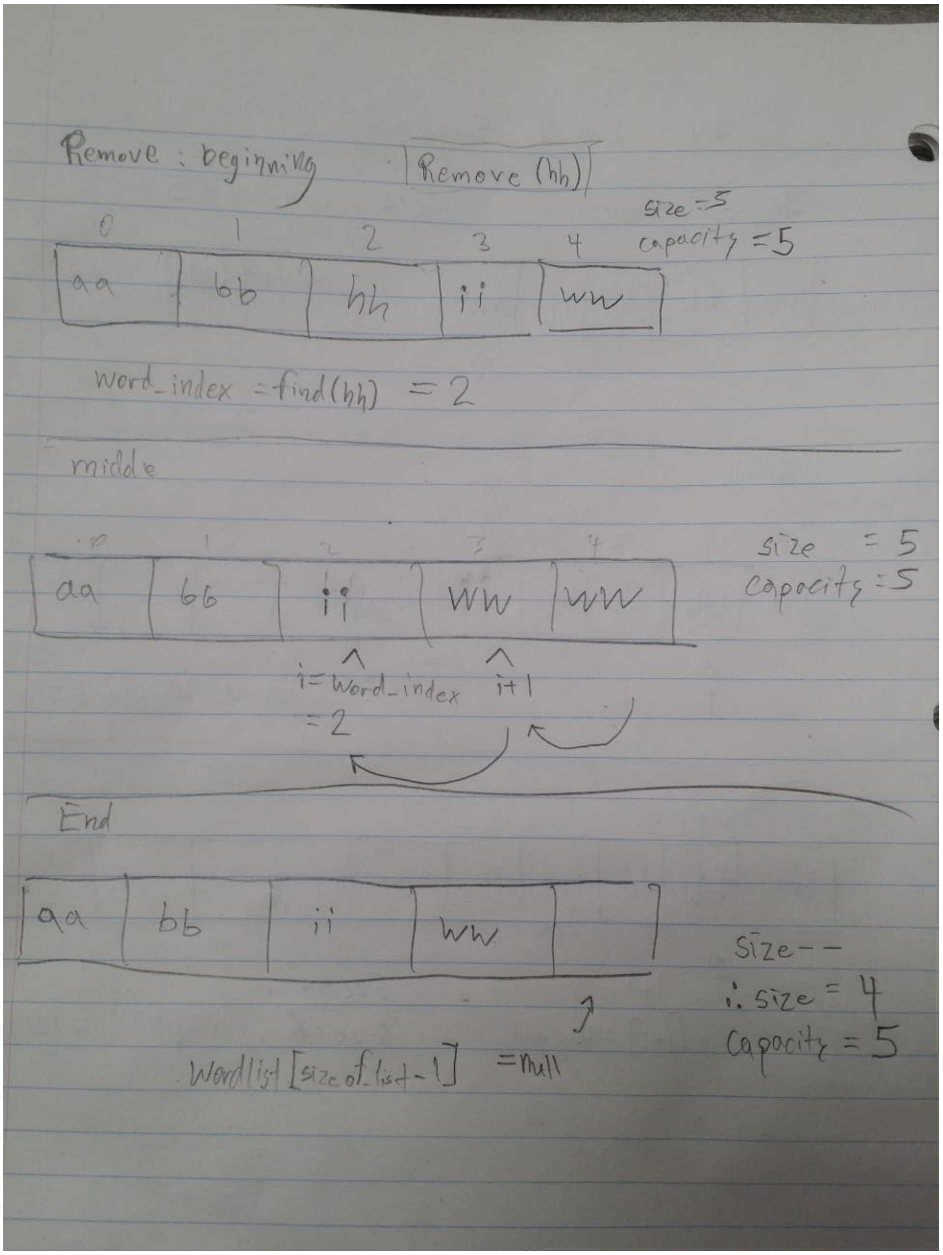
aa	bb	dd	hh	ii	ww				
	^ i	^ i+1							

[i+1] =newword

size++

size=6, capacity=10

Above : representation of method Insert(), using Insert(dd) on a full list



Remove : beginning

Remove (hh)

size = 5
capacity = 5

0	1	2	3	4
aa	bb	hh	ii	ww

word_index = find(hh) = 2

middle

0	1	2	3	4
aa	bb	ii	ww	ww

size = 5
capacity = 5

i = word_index = 2

i+1

End

aa	bb	ii	ww	

size--
∴ size = 4
capacity = 5

Wordlist [size of list - 1] = null

Above: Representation of remove(), using remove(hh) on a list containing hh