Huzaifa Mirza – mirzahw, 1324186

Lab 1 – Report

2SI4

Dates

- Started: 19/01/2015 – Monday 19, 2015
- Halfway: 21/01/2015 – Wednesday 21, 2015
- Finished: 22/01/2015 – Thursday 22, 2015

Sources

- I searched the compareTo function online
  - http://www.tutorialspoint.com/java/java_string_compareto.htm
  - http://www.tutorialspoint.com/java/number_compareto.htm

Description of Algorithms

- **insert():**
  - The purpose of this method is to add a new word into an already existing array of words. This is done in many steps. First this method checks to see if the array is already at maximum capacity, cap == size. This means that there isn't any space to add a new word. If the array is full then this method creates a new array with double the capacity. The method then transfers all the words from the smaller array to the bigger one and set that as the main array, the array all the work is done with. Once insert() is sure that the new word can be added in the main array it proceeds to check where the new word should be added, the index of the new word. It does this by using a for loop, starting from the beginning and going through each word and checking if the word to be inserted is smaller than the ith word in the array. Once the word to be inserted is smaller than the array word the for loop stops and the variable i, used for for loop, is the index the word should be inserted. Once is has the index the new word should be inserted, it checks to see if that word already exists there using an if statement and the java code compareTo(). If the word already exists in the list then insert() just returns the array. Otherwise, insert() moves all the words after the ith index, inclusive, forwards. This leaves the ith index empty so the word to be inserted is just added in that index.
  - If we have a list with 5 words, String List[] = {"arrow", "carrot", "ear", "goat", "jello"}

    List = {"arrow", "carrot", "ear", "goat", "jello"}
    Size = 5 Capacity = 10
    insert("hat")
    output would be –
    {"arrow", "carrot", "ear", "goat", "hat", "jello"}
    Size = 6 Capacity = 10

- **remove():**
  - The purpose of this method is to remove a word from an array of words. First what this method does is it makes sure that the array has at least one word in it using an if statement checking if size > 0. Once remove() is sure that the array is not empty, it calls the find method to check if the word is in the array. If the word is in the array then remove() moves all the words in the array that come after that word back one index. This cause the word we want to remove to be over written by the word that comes after it in the array. After remove() has moved all the words back one index is subtracts one from size.
  - If we have the same list, String List[] = {“arrow”, “carrot”, “ear”, “goat”, “jello”}

    List = {“arrow”, “carrot”, “ear”, “goat”, “jello”}
    Size = 5 Capacity = 10
    remove(“ear”)
    output would be –
    {“arrow”, “carrot”, “goat”, “jello”}
    Size = 4 Capacity = 10