AMY JIANG

HOMEWORK# 4

CS/IS 211

MARCH 23, 2017

Pages 114-115 exercises (C++ or psuedocode are both ok!  Remember though UML is not pseudocode you need to give ideas for the implementation in your pseudocode see my binarySearch pseudocode in Chapter 2 if you're stuck on how to do the pseudocode)

#8 (15 points) - Note:  this should be a function we could add to the ArrayBag (not the recursive or the resizable one the regular one) that would just remove a random entry from the bag.  So it removes any random entry.

/\*\* 8. Specify and define a method for ArrayBag that removes a random entry from the bag\*\*/

template<class ItemType>

bool ArrayBag<ItemType>::removeRandom() {

srand (time(NULL));

//cout << "The current size is " << getCurrentSize() << endl;

bool canRemoveItem = !isEmpty(); //if false don't remove item since bag is EMPTY.

if(canRemoveItem) {

int value = getCurrentSize() - 1; //must minus one because index starts at 0!

if (value <= 0) { //CASE: when there's only one value in the bag remove that value

itemCount--;

items[0] = items[itemCount];

remove(0);

canRemoveItem = true;

}

else {

int randEntryToRemove = (rand() % value) ; //range from 0 to value;

cout << "The random value is " << randEntryToRemove << endl;

itemCount--;

items[randEntryToRemove] = items[itemCount];

remove(randEntryToRemove);

canRemoveItem = true;

}

}

return canRemoveItem;

}

#9 (15 points)  - Same note as for #8.  This should be a function we could add to the ArrayBag  (not the recurisve or resizable one, the regular one).  It should be a constructor that takes an array as an argument and creates an ArrayBag.  You can assume the array size is less than the maximum size allowed for the bag.

template<class ItemType>

ArrayBag<ItemType>::ArrayBag(const ItemType\* arr, int size) {

int value = size/sizeof(ItemType);

cout << "The size of my array is " << value << endl;

for (int i = 0 ; i < value; ++i) {

items[i] = arr[i];

itemCount++;

}

}