Amy Jiang

March 5, 2017

CS/IS 211

+intersectBags (oldBag1 : Bag ) : Bag

***A***. Write specifications for a new intersectBags method described above. Include a statement of  purpose, the preconditions, a description of the arguments, and a description of any return value.

Statement of Purpose: Check two bags to see if any of the entries match. For those that match, return a creation of a new bag that only includes the values that are matched (include duplicates).

Preconditions:

* Bag receiving the call must not be empty
* **oldBag1** must be the same data type/object
* Check if any bag is empty; if so, return an empty bag (since there will be no intersections)

Postconditions:

* **newBag** is the Bag returned which is the bag formed from **oldBag1** and bag receiving the call

Description of arguments:

* **Bag oldBag1** is a bag, which contains a random number of objects; these objects can contain duplicates.

Description of return values:

* Return **Bag newBag**, which is a bag that is formed from the intersections of oldBag1 and receiving call.

***B.*** Pseudocode implementation

Template<class ItemType>

Bag<ItemType> Bag<itemType>::intersectBags( const Bag<itemType>& oldBag1) {

Bag<itemType> newBag; //declare newBag that will be returned @ end

if(oldBag1.isEmpty()) {

//make sure the bag receiving the call isn’t empty

//return an empty newBag if oldBag is empty

return newBag.clear();

}

else {

//create a for loop that checks if oldBag1 intersects object(s) of bag receiving the call

//check if there is a duplicate

for( int i = 0; i < oldBag1.getCurrentSize(); ++i) {

for(int k = 0; k < getCurrentSize(); ++k) {

//use the contains to check if the bag receiving the call intersects w/ oldBag1

//if true, add the matched object to newBag newBag.add(//matched value)

}

}

return newBag;

}

}