

Homework #2  
Due Date is in Moodle  
30 pts

This question is based on an exercise on Page 28 of your book

1. 30 points

The *intersection* of two bags is a new bag containing the entries that occur in both of the original two bags. Design and specify a method **intersection** for the ADT bag that returns as a new bag the intersection of the bag receiving the call to the method and the bag that is the method's one argument. Include sufficient comments to fully specify the method.

Note that the intersection of two bags might contain duplicate items. For example, if object x occurs five times in one bag and twice in another, the intersection of these bags contain x two times. Specifically, suppose that bag1 and bag2 are bags; bag1 contains the strings a, b, and c; and bag2 contains the strings b, b, d, and e. The expression bag1.intersection(bag2) returns a bag containing only the string b. Note that intersection does not affect the contents of bag1 and bag2.

**HINT:** You will need to read BagInterface.h and use the bag methods to make this solution easier.

- 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.
- b. Write a pseudocode implementation of this method (or do something like I started below with a sort method sort of combine pseudocode and C++). It doesn't have to be perfect pseudocode but in general it should show how this method will work (I will be very lenient on pseudocode). You can write it in C++ but you don't have to, the reason I didn't require it in C++ is because I know especially for the CSIS 135 people the pseudocode might be difficult.

***Example pseudocode for a public method called sort to sort an array member variable in a class. This is just an example – but notice I sort of mixed C++ with pseudocode – this is what you can do for your intersect method for your bag:***

```
void sort(int arr[], int size)
{
    for (int j = 0; j < size; j++)
    {
        if arr[i] > arr[j]
            swap arr[i] and arr[j]
        etc etc...note this is not the correct implementation for sort.
    }
}
```