Table 1. The time complexities of the three methods

	ResizeableArrayBag			LinkedBag		
	union	**intersection**	**difference**	**union**	**intersection**	**difference**
Time Complexity in the Best Case	O(n)	O(n)	O(n)	O(n)	O(n)	O(n)
Time Complexity in the Worse Case	O(n^2)	O(n^2)	O(n^2)	O(n^2)	O(n^2)	O(n^2)

ResizeableArrayBag

Union

Best Case: O(n); The best case for union is when it just needs to add n elements of one bag to the result bag.

Worse Case: O(n^2); The worst case occurs when the algorithm has to iterate through n elements of the first bag and add it to the result bag. Then it has to go through n elements of the second bag and add it to the result bag, in O(n^2) time.

Intersection

Best Case: O(n); The best case occurs when all n elements of both bags are the same so the method is basically adding the each element to the result bag. Since it only iterates through n elements, the time complexity is O(1).

Worse Case: O(n^2); The worst case is when the elements of both bags are unique so that the algorithm has to iterate through n elements for one bag and another n elements for the other bag, which is done in O(n^2) time.

Difference

Best Case: O(n); The best case is when the elements are unique for both bags and everything is added to the result bag. The algorithm only has to iterate through n elements to get the contents and add it to the result bag.

Worse Case: O(n^2); The worst case occurs when the elements of both bags are the same and the algorithm has to go through n elements to get the contents of each item. Then it has to check whether it is already in the other bag and find the difference, which is done in O(n^2) time.

LinkedBag

Union

Best Case: O(n); The best case occurs when it just needs to add n elements to the result bag, performed in O(n) time.

Worse Case: O(n^2); The worst case occurs when the algorithm has to go through n elements of one bag and another n elements of the other bag, which in the end is in O(n^2) time.

Intersection

Best Case: O(n); The best case occurs when the two bags are the same and the algorithm has to just go through n elements to get the contents of the bag and add it to the result bag, which is done in O(n) time.

Worse Case: O(n^2); The worst case occurs when both bags are entirely different and has to go through n elements to get the contents. Then it iterates through another n elements to compare them to find the common elements, which in the end is O(n^2).

Difference

Best Case: O(n); The best case occurs when the elements of both bags are unique, hence all the elements are added to the result bag. The algorithm only iterates n times to get the contents of the bags, done in O(n) time.

Worse Case: O(n^2); The worst case occurs when the elements of both bags are the same and there are no differences in the bags. The algorithm still has to iterate through n elements to get the contents and goes through it again to compare and calculate the difference, performed in O(n^2).