

ResizableArrayBag:

* Union
  + Best: O(n)
  + Worst: O(n)
    - Both cases would be O(n) because no matter what, the time it would take to complete the program would depend on the number of terms there are in each bag. The fewer entries there are, the faster the code would be (Best case), and the more entries there are, the longer it would take (Worst case).
* Intersection
  + Best: O(n)
  + Worst: O(n)
    - Both cases would be O(n) because in both cases, the time it would take to complete the program depends on how many entries are in each bag. After checking the number of entries, the program looks at one entry in one back then checks to see if the other bag contains it. It then adds the entry that occurs in both bags to bag one. The equations would be O(2n +2) which would then change to O(n).

* Difference
  + Best: O(n)
  + Worst: O(n)
    - Much like intersection, for both cases, the time complexity would be O(n). This is because the program first checks the amount of entries. Then it compares each entry in the first bag to the second bag. Because the program is just taking away from a bag the time complexity depends on the amount of entries that must be compared.

LinkedBag:

* Union
  + Best: O(n)
  + Worst: O(n)
    - Both cases for the union method in LinkedBag are O(n). This is, because the program is just adding a node after other nodes. It is linear with the input because it does use “for” loops. The fewer the nodes that need to be imputed from bag 2 to bag 1, the less times it needs to run the “for” loop.
* Intersection
  + Best: O(n^2)
  + Worst: O(n^2)
    - In both cases, the time complexity would be O(n^2). This is because no matter what the case, each entry would have to go through two for loops before anything could happen to the returned bag. First the program checks the number of entries, then it gets the size of the array. After it does both of those, the arrays are compared. The full equation would be O(n \*n +1) = O(n^2).

* Difference
  + Best: O(n^2)
  + Worst: O(n^2)
    - Both cases for the difference method in LinkedBag are O(n^2) because the program has to run nested for loops. These for loops - alone - run linearly with the length of the bag. Once you add another for loop inside of the for loop (nested) it would need to do the length of the bag\*length of the bag which, if we put them into variables, would be O((n\*n) + 1) (plus 1 because of the if) which is also O(n^2).