Problem 3)

1. This search algorithm achieves O(logN) time because it works as a binary search. What this means is that once you have a list of numbers, this type of search splits it in half first and checks whether the number you are looking for is to the left or the right of the current value at the half. If it is less than, it will be to the left and if it is greater than it will look to the right. Then it will do this over and over again until it finds the element or the element is not in the array. Because of this splitting, it gives the algorithm a time complexity of logN.