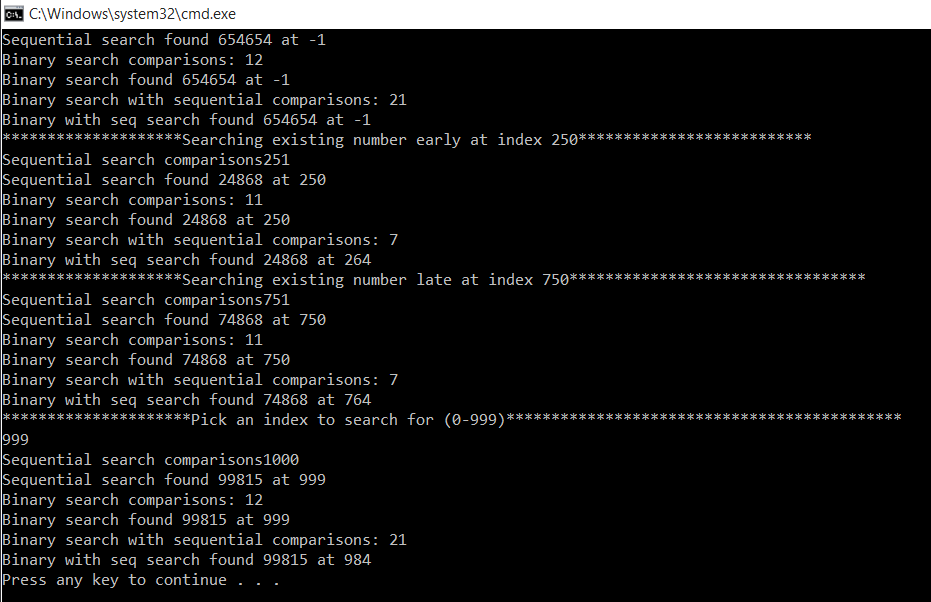
Problem 1 report:



A discussion of your results with screenshots of different test cases.

I chose four cases

one which was user input for number search that ended up false

one from the lower quartile, always came up true

one from the upper quartile, always came up true

one user chosen from an index, always came up true

A discussion of the time complexity of the Binary Search Algorithm and Sequential Search Algorithm in terms of Big O notation.

Binary search time complexity: O(log2(n))

The search bisects until the last unit one giving log base 2 of n as the time complexity

Sequential search time complexity: O(n)

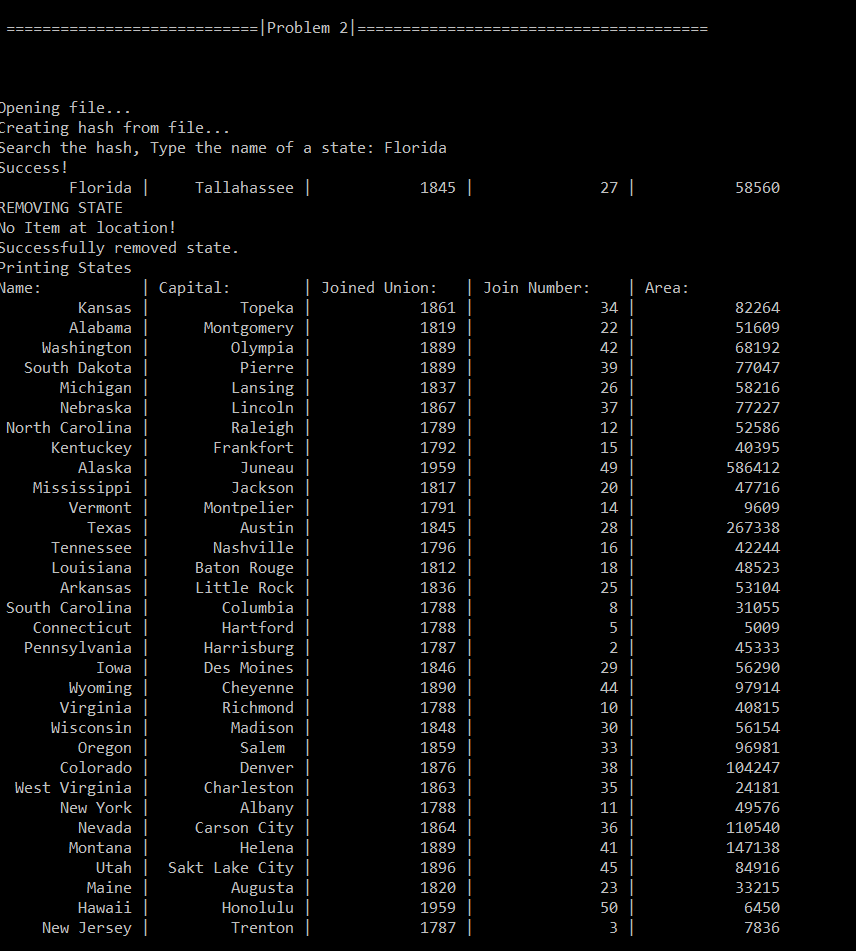
The search at worst case would do equal comparisons to length of list

A discussion of the memory efficiency of the Binary Search Algorithm and Sequential Search Algorithm.

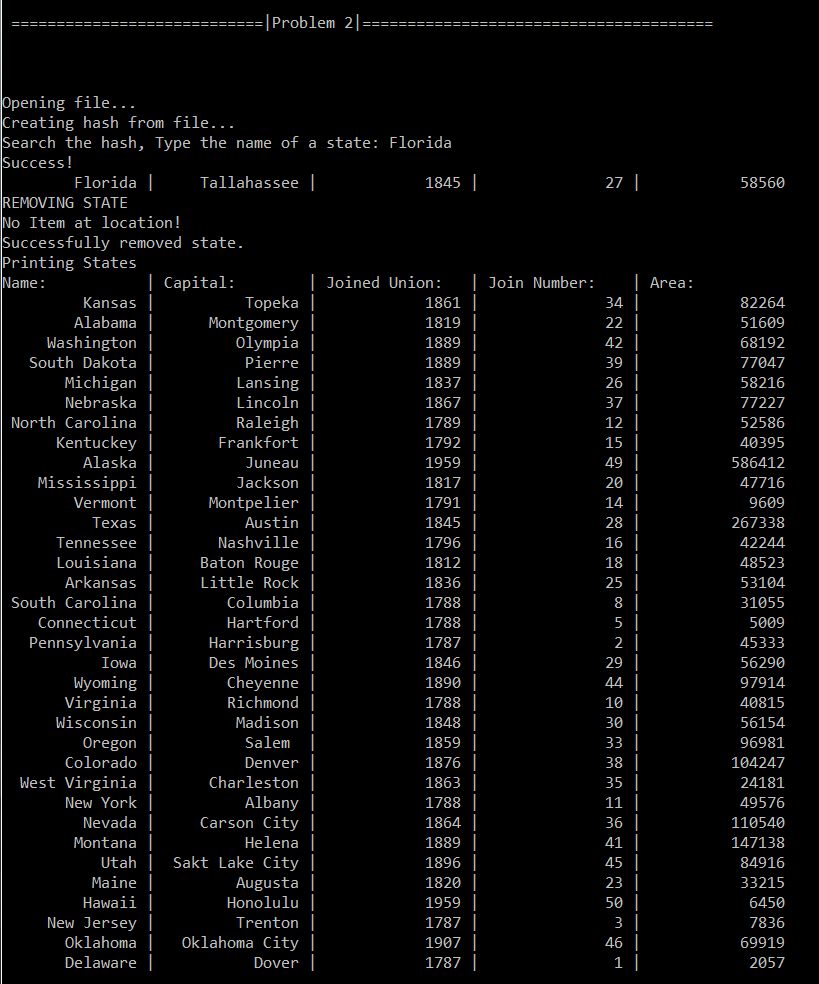
They both have a memory efficiency of O(1) since they do not require any additional memory for the function

Problem 2

Case 1 Not found



Case 2 Found and removed



 A discussion of your results with screenshots of these results.

I resulted in a successful state hash with one case above where a fake state was given and not found and a real state was given and was removed

 A discussion of the time complexity of the Hashing Algorithm in terms of Big O notation.

The time complexity of the hashing algorithm in worst case is O(n) where the worst case is that the hash does not exist. The average case is O(1) where having a big enough table will allow the program to search the item right away, regardless of data size.

 A discussion of the memory efficiency of the Hashing Algorithm.

The memory efficiency is O(2n) because there are two tables that hold data for the hash.