Let \( n \) be the count of numbers in a collection of base10 numbers. Suppose zero is the minimum number and \( k \) is the maximum number in the collection. The time complexity of counting sort is

This is off the top of my head...

You have to process the array of \( n \) numbers, so that takes \( \theta(n) \) time. Then you have to process the counts array, which takes \( \theta(k) \) time. The overall time depends on which is bigger, \( n \) or \( k \).

Thank you!