**Module 5 CTA – Radix Sort Algorithm Analysis**

Michael Masters

Colorado State University Global

CSC400: Data Structures & Algorithms

Dr. Shaher Daoud

January 19, 2025

**Module 5 CTA – Radix Sort Algorithm Analysis**

Radix Sort is a non-comparative sorting algorithm that works by sorting the elements of an integer or string array based on the number of significant digits (radix) contained within the elements. The algorithm can be implemented starting with the least significant digit (ascending) or the most significant digit (descending). The result is a sorted array that is achieved without comparing elements.

**Evaluation of Radix Example**

To illustrate how Radix is executed, consider an integer array containing the elements [783, 99, 472, 182, 264, 543, 356, 295, 692, 491, 94]. The array will be sorted based on the least significant digit (ascending).

1. Sort numbers based on last (right most) digit:
   1. [{491}, {472, 182, 692}, {783, 543}, {264, 94}, {295}, {356}, {99}].
2. Sort based on next left digit:
   1. [{543}, {356}, {264}, {472}, {182, 783}, {491, 692, 94, 295, 99}]
3. Sort on left most digit:
   1. [94, 99, 182, 264, 295, 356, 472, 491, 543, 692, 783]

**Big-O Complexity**

The time/space complexity (aka Big-O notation) of this algorithm is , where is the number of elements in the array, and is the largest value in the array. This represents linear complexity in relation to both the array size and the magnitude of the largest value. For each of the elements, the algorithm performs operations. As a result, the total number of operations scales proportionally to .