Cory McDonald
4/9/2015
Algorithms

Program #10 Report

1. Radix Sort With Counting Sort

- How did you implement it?
- Took each column from right to left and added the digits into buckets, then added all the entries in that array and copied the values to the returnMatrix. Then assigned the original matrix to be the almost sorted array.
- Data structures: Array
- Important variables: bucket, returnMatrix,and the max number of digits that exist in the input

2. Radix Sort With Linked Lists

- How did you implement it?
  For this one I used a 10 linked list ranging from 0 to 9 to use as my buckets, from there I called two functions, distribute and coalesce. The distribute function puts everything into buckets and the coalesce combines all the linked list
- Data structures: Linked List, passed in an Object[] array for the original numbers
- Important variables: masterlist, q, radix.

3. Comparing Times

- Which one typically ran faster?
  The counting sort version ran faster.
- Why is that?
  Although the array has memory overhead it is much faster at accessing the sequential elements.
- Was there a time that the algorithm you did not expect to be faster was faster?
  Honestly I thought the linked list version would be faster because I just viewed arrays has very simple, but for radix sort it seems like doing something simple is much faster because of the reduced computational time.