CSCI 102 assignment 8 – Radix sort

April 14

Say we want to sort the numbers $[456, 132, 435, 57, 432, 786, 101]$. In quicksort and mergesort, we saw the benefit of splitting the problem into smaller problems. In this assignment you will be implementing a version of a sorting algorithm, radix sort, which uses the same idea.

For the array above, we note that in the sorted list, all the numbers with 7 as their hundreds digit will be after those with 4 as their hundreds digit, which in turn will be after those with 100. So, we split the sorting problem into 10 problems – sorting 10 "buckets" each determined by their hundreds digit:

$$[[\underline{0}57], [\underline{1}32, \underline{1}01], [\underline{4}56, \underline{4}35, \underline{4}32], [\underline{7}86]]$$

Now if we can sort each "bucket", then we can concatenate the sorted buckets to get the total sorted list.

First note that if a bucket has only one entry, then it is already sorted. To sort a bucket with more than one entry, we can simply apply radix sort again to the next digit, which in this case is the 10s digit. Thus, we break the bucket $[456, 435, 432]$ into

$$[[4\underline{3}5, 4\underline{3}2], [4\underline{5}6]].$$

Finally we can break $[4\underline{3}5, 4\underline{3}2]$ into $[[43\underline{2}], [43\underline{6}]]$.

In this example we have sorted numbers with up to 3 digits. If however we added a number like 1035 to our list, we would need to perform an extra first step to split our problem up by the thousands digit.

- Implement the radix sort described above in a recursive algorithm `Integer[] radixSort(Integer[] arr)`. This assignment is open ended; implement the method using your best judgment! Some advice: It should be able to handle integers that may have many digits. You will need an auxiliary variable counting which digit you are on. You may use whatever structure you like for the buckets, but I recommend creating 10 buckets (`Integer` arrays) every time you call the sorting method (don't worry about using too much memory when writing this). To get the k-th digit of a number you can either convert the number to a string or perform integer division by 10^{k-1} and then modulo by 10.
- Use your implementation to sort $[4456, 13542, 4315, 547, 42, 0, 12310, 4452, 10]$.
- If n is the number of elements in a list, and ℓ is the maximum number of digits in each entry of the list, what is the asymptotic worst case complexity of this algorithm? In class we gave a proof that the most efficient possible algorithm for sorting is $O(n \log n)$; how does radix sort compare? Please **explain**!
- Describe very briefly how one could use radix sort to sort strings.

Please submit your code and answers to the questions in a zipped folder on Brightspace by April 21.