CS 240: Data Structures and Data Management **Spring 2023**

Tutorial 06: June 19

1. **Runtime of MSD-RadixSort**
   Analyze and justify runtime of MSD-RadixSort. As discussed in the lecture, the run-time of MSD-RadixSort is $\Theta(mnR)$ where R is the base of each element, m is the number of digits, and n is the number of elements.

```
MSD-Radix-sort(A, l, r, d)
    if l < r
        bucket-sort(A[l,r], d)
        if there are digits left    // recures in sub-arrays
            l' = l
            while (l' < r) do
                r' = maximal s.t. A[l',...,r'] all have the same dth digit
                MSD-Radix-sort(A, l', r', d+1)
                l' = r'+1
```

2. **Numbers in Range**
   We have an array A of n non-negative integers such that each integer is less than k. Give an $O(n+k)$ time preprocessing algorithm such that queries of the form "how many integers are there in A that are in the range $[a, b]$?" can be answered in $O(1)$ time. Note that a and b are not fixed; they are parameters given to the query algorithm.

3. **Multiplicity Sorting** Consider the problem of sorting an array A of n elements with multiplicity n/k. That is, A consists of k distinct elements $(y_1, y_2, \ldots, y_k)$, where each y_i occurs n/k times in A. Prove that any algorithm in the comparison model requires $\Omega(n \log k)$ comparisons to sort A in the worst-case.
   Note: $\forall m \geq 0, \left(\frac{m}{e}\right)^m \leq m! \leq m^m$.