# ITS202: Algorithms and Data Structures

## String Sorts

Ms. Sonam Wangmo

Gyalpozhing College of Information Technology
Royal University of Bhutan

October 20, 2020

# Radix Sort

We consider two fundamentally different approaches to string sorting.

1. LSD referred to as least-significant-digit (LSD) string sorts.
2. MSD referred to as most-significant-digit (MSD) string sorts.

# LSD radix sort

## Concepts

- Consider characters from right to left.
- Sort using dth character as the key (Using key-indexed counting)
- LSD sorts fixed length strings in ascending order.
- Sorting such string can be done using key-indexed counting.
- If the strings are each of length W, we sort the strings W times with key-indexed counting, using each of the positions as the key, proceeding from right to left.

# LSD radix sort

```
1   public class LSD
2       {
3           public static void sort(String[] a, int
                W)
4           {
5               //Sort a[] on leading W characters.
6               int N = a.length;
7               int R = 256;
8               String[] aux = new String[N];
9               for (int d = W-1; d >= 0; d--)
10              {
11              // Sort by key-indexed counting on
                    dth char.
12                  int[] count = new int[R+1];
13              // Compute frequency counts.
14                  for (int i = 0; i < N; i++)
```

# LSD radix sort

```
1                       count[a[i].charAt(d) + 1]++;
2                       for (int r = 0; r < R; r++)
3                       // Transform counts to indices.
4                       count[r+1] += count[r];
5                       for (int i = 0; i < N; i++)
6                       // Distribute.
7                       aux[count[a[i].charAt(d)]++] =
                            a[i];
8                       for (int i = 0; i < N; i++)
9                       // Copy back.
10                      a[i] = aux[i];
11                  }
12              }
13          }
```

# LSD radix sort

To sort an array a[] of strings that each have exactly W characters, we do W key-indexed counting sorts: one for each character position, proceeding from right to left.

# LSD radix sort

| input (W = 7) | d = 6 | d = 5 | d = 4 | d = 3 | d = 2 | d = 1 | d = 0 | output |
|---|---|---|---|---|---|---|---|---|
| 4PGC938 | 2IYE230 | 3CIO720 | 2IYE230 | 2RLA629 | 1ICK750 | 3ATW723 | 1ICK750 | 1ICK750 |
| 2IYE230 | 3CIO720 | 3CIO720 | 4JZY524 | 2RLA629 | 1ICK750 | 3CIO720 | 1ICK750 | 1ICK750 |
| 3CIO720 | 1ICK750 | 3ATW723 | 2RLA629 | 4PGC938 | 4PGC938 | 3CIO720 | 1OHV845 | 1OHV845 |
| 1ICK750 | 1ICK750 | 4JZY524 | 2RLA629 | 2IYE230 | 1OHV845 | 1ICK750 | 1OHV845 | 1OHV845 |
| 1OHV845 | 3CIO720 | 2RLA629 | 3CIO720 | 1ICK750 | 1OHV845 | 1ICK750 | 1OHV845 | 1OHV845 |
| 4JZY524 | 3ATW723 | 2RLA629 | 3CIO720 | 1ICK750 | 1OHV845 | 2IYE230 | 2IYE230 | 2IYE230 |
| 1ICK750 | 4JZY524 | 2IYE230 | 3ATW723 | 3CIO720 | 3CIO720 | 4JZY524 | 2RLA629 | 2RLA629 |
| 3CIO720 | 1OHV845 | 4PGC938 | 1ICK750 | 3CIO720 | 3CIO720 | 1OHV845 | 2RLA629 | 2RLA629 |
| 1OHV845 | 1OHV845 | 1OHV845 | 1ICK750 | 1OHV845 | 2RLA629 | 1OHV845 | 3ATW723 | 3ATW723 |
| 1OHV845 | 1OHV845 | 1OHV845 | 1OHV845 | 1OHV845 | 2RLA629 | 1OHV845 | 3CIO720 | 3CIO720 |
| 2RLA629 | 4PGC938 | 1OHV845 | 1OHV845 | 1OHV845 | 3ATW723 | 4PGC938 | 3CIO720 | 3CIO720 |
| 2RLA629 | 2RLA629 | 1ICK750 | 1OHV845 | 3ATW723 | 2IYE230 | 2RLA629 | 4JZY524 | 4JZY524 |
| 3ATW723 | 2RLA629 | 1ICK750 | 4PGC938 | 4JZY524 | 4JZY524 | 2RLA629 | 4PGC938 | 4PGC938 |

Figure 1: LSD

# LSD radix sort

Q. What if strings are not all of same length?

# LSD radix sort

Q. What if strings are not all of same length?

# MSD radix sort

## Concepts

- Consider characters from left to right order.
- Sort using dth character as the key (Using key-indexed counting)
- A general-purpose string sort, where strings are not necessarily all the same length
- Sorting such string can be done using key-indexed counting.
- If the strings are each of length W, we sort the strings W times with key-indexed counting, using each of the positions as the key, proceeding from right to left.

# MSD radix sort


Figure 2