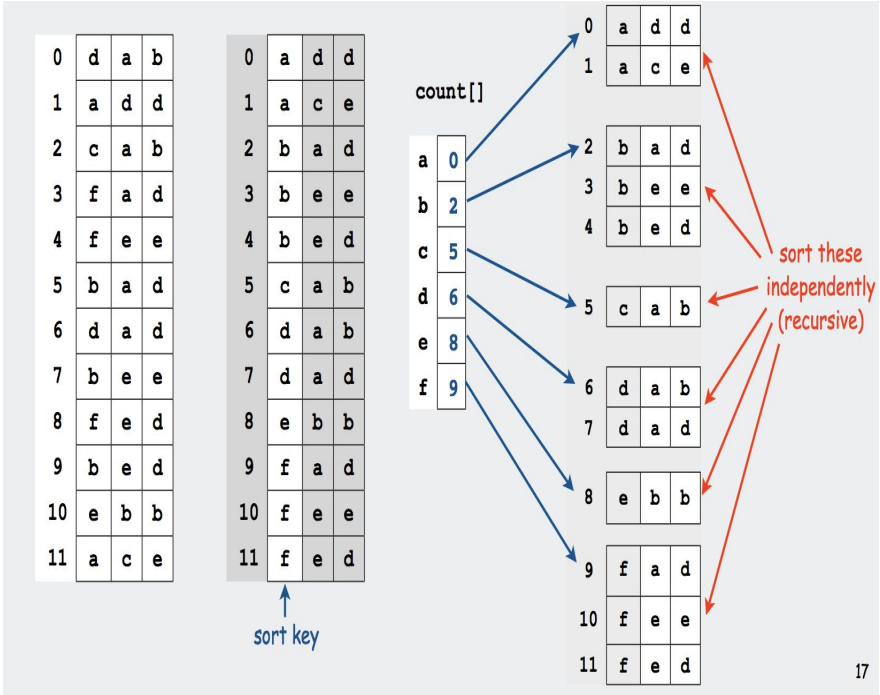
# MSD Radix Sort

Lindsey Chapman
Tae Yeon Kim

# What is MSD Radix Sort?

- Most Significant Digit (MSD) radix sort
- Redistributes digits in left to right order at the leftmost digit into buckets
- Recursion occurs within the buckets to further sort
- Key indexed counting and bucket sort rather than comparisons



# Code Implementation

```
public static void msd(String[] a)
{  msd(a, 0, a.length, 0);  }

private static void msd(String[] a, int lo, int hi, int d)
{
   if (hi <= lo + 1) return;
   int[] count = new int[256+1];
   for (int i = 0; i < N; i++)
      count[a[i].charAt(d) + 1]++;
   for (int k = 1; k < 256; k++)
      count[k] += count[k-1];
   for (int i = 0; i < N; i++)
      temp[count[a[i].charAt(d)]++] = a[i];
   for (int i = 0; i < N; i++)
      a[i] = temp[i];
   for (int i = 0; i < 255; i++)
      msd(a, l + count[i], l + count[i+1], d+1);
}
```

key-indexed counting

count frequencies

compute cumulates

move records

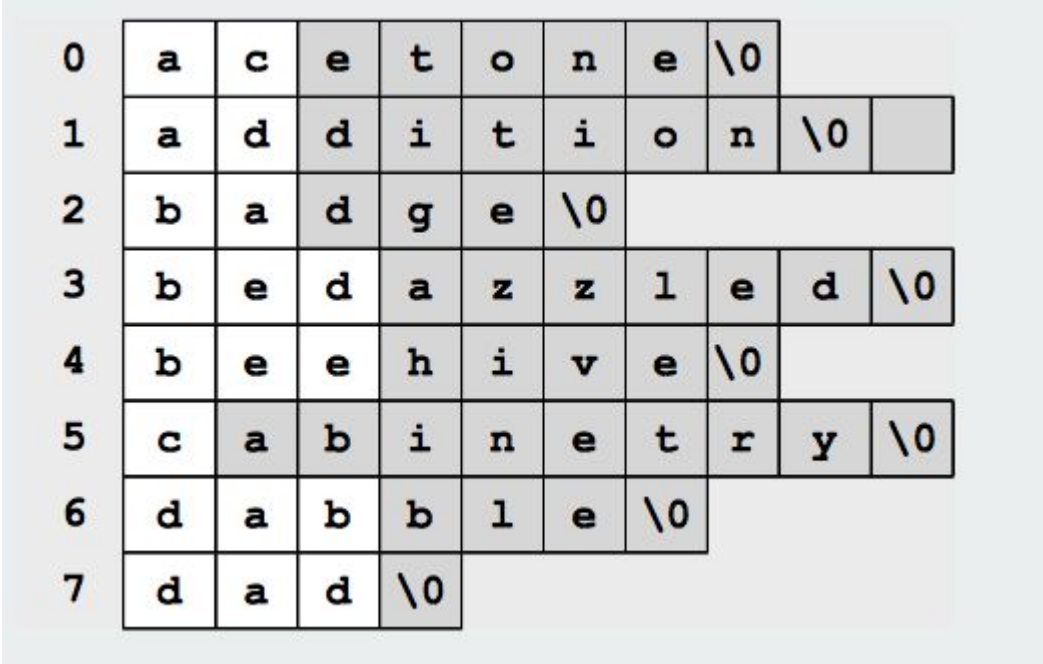
copy back

# MSD vs LSD

- When working with integers of fixed length, MSD is more efficient than LSD because it may not have to examine every digit of each integer

# MSD vs LSD

- MSD can be used to sort strings of variable length, unlike LSD



# MSD vs LSD

- MSD uses recursion, so it requires more space than LSD
  - This means that MSD is much slower than LSD when working with small files--however, we can switch to another sorting algorithm for small values of N
- LSD has to be stable in order to work correctly, but MSD can either be made stable or unstable

# Complexity Comparisons

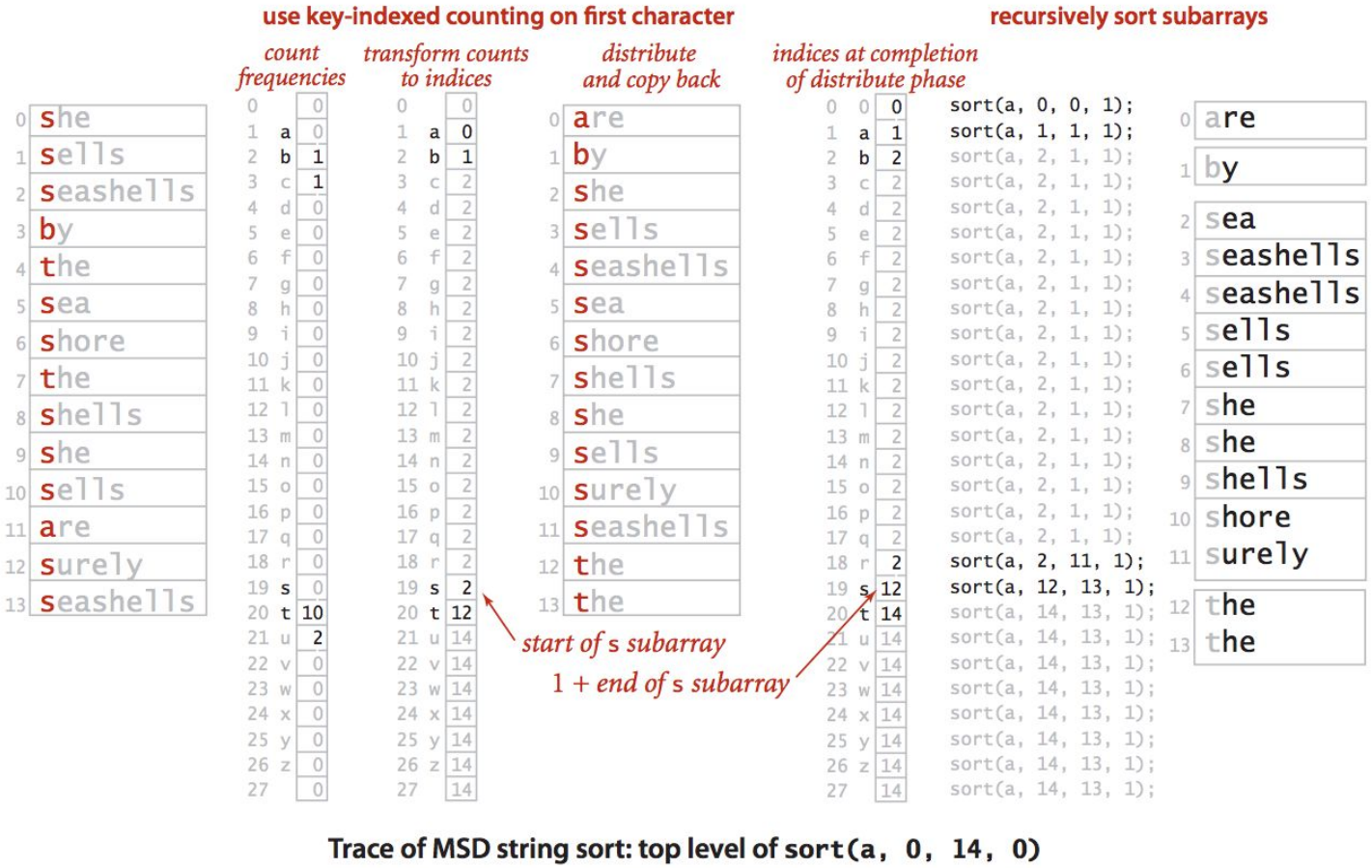
| | stable? | inplace? | run time? | extra space | sweet spot |
|---|---|---|---|---|---|
| *LSD string sort* | yes | no | O(NM) | O(N + B) | short fixed-length strings |
| *MSD string sort* | sometimes | no | Between O(N) and O(Nm) | O(N + MB) | random strings |

N = number of strings to sort
R = number of characters in alphabet
m = average length of string
M = max length of string

# Time and Space Complexity of MSD

- Best case time complexity is O(N) and worst case time complexity is O(Nm), where N is the number of entries and m is the average length of the strings
  - LSD has time complexity O(NM) in both the best and worst case scenario, where M is the fixed length of all the strings
- MSD requires O(N + MB), where B is the size of the radix
  - The size of the radix corresponds to the range of values that a given type can take--for example, there are 256 = B possible characters in a string, but only 10 possible characters in an integer in base 10, and there are 2 possible characters in binary
  - LSD requires only O(N + B) extra space

# Example




Trace of MSD string sort: top level of `sort(a, 0, 14, 0)`

# Citations

M. T. Goodrich, R. Tamassia. *Data Structures and Algorithms in Java, 6th edition.* Wiley, 2014. Print.

Sedgewick, Wayne. *Algorithms (4th edition)*. Boston: Addison-Wesley Professional, 2011. Print.