Lecture 14
Radix Sort

Department of Computer Science
Hofstra University

Radix and Radix Sort

- Radix = “The base of a number system”
- Radix is another term of “base” : number of unique digits, including the digit zero, used to represent numbers
- Radix of numbers:
  - Binary numbers have a radix of 2
  - decimals have a radix of 10
  - hexadecimals have a radix of 16

Radix and Radix Sort

- Radix sort was first used in 1890 U.S. census by Hollerith
- Very efficient when sorting a large number of elements
  - O(n*k). n: number of elements; k: number of digits in the largest number
- May use more space than other sorting algorithms
  - E.g., bubble sort is in-place sorting.
- Basic idea: Bucket sort on each digit, from least significant digit to most significant digit.

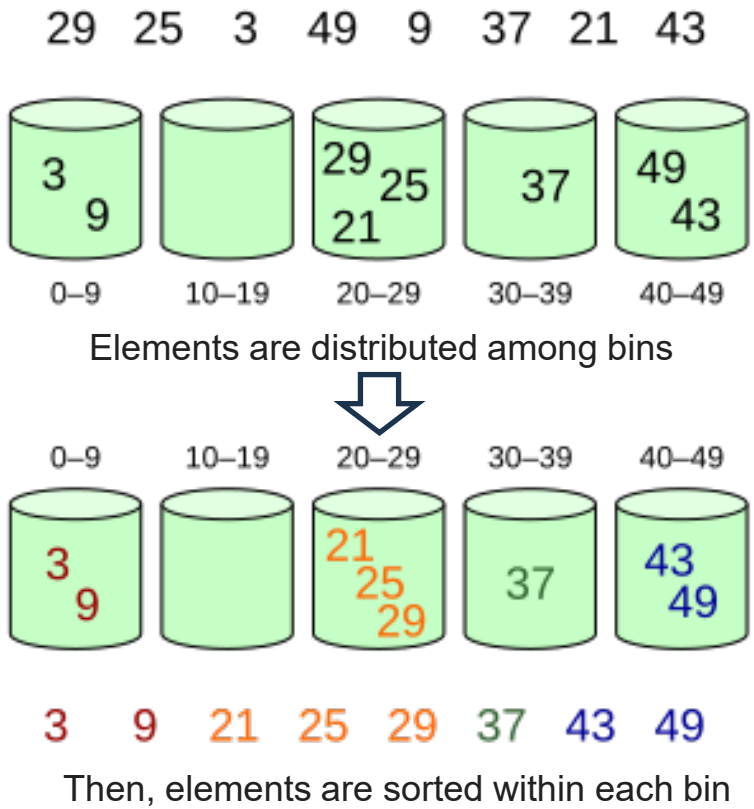
Radix Sort Algorithm

```
radix_sort(A, n, k) {
   /* A: array; n: number of elements; k: number of digits in the largest number
*/
   create buckets  (buckets can be arrays or lists)
        for (d = 0; d <k; d++) {
                /* sort A using digit position d as the key. */
                for (i = 0; i<n; i++) {
                        p = the d-th digit  (from right) of A[i]
                        Add A[i] to bucket p
         }
            A = Join the buckets
   }
}
```

Time complexity O(n*k)

Bucket Sort

- Bucket sort is a comparison sort algorithm that works by distributing the elements of an array into a number of buckets and then each bucket is sorted individually using a stable sorting algorithm, e.g., Insertion Sort or Merge Sort.
- This algorithm is efficient when the input is uniformly distributed over a range.

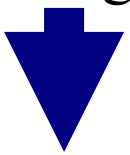


Bucket Sort | GeeksforGeeks
https://www.youtube.com/watch?v=VuXbEb5ywrU

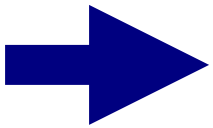
https://en.wikipedia.org/wiki/Bucket_sort

Bucket Sort as used in Radix Sort

- Use bucket array of size R for radix of R
- Put elements into the correct bucket in the array
- R = 5; unique digits (0,1,2,3,4); list = (0,1,3,4,3,2,1,1,0,4,0)



Buckets	
= 0	0,0,0
= 1	1,1,1
= 2	2
= 3	3,3
= 4	4,4



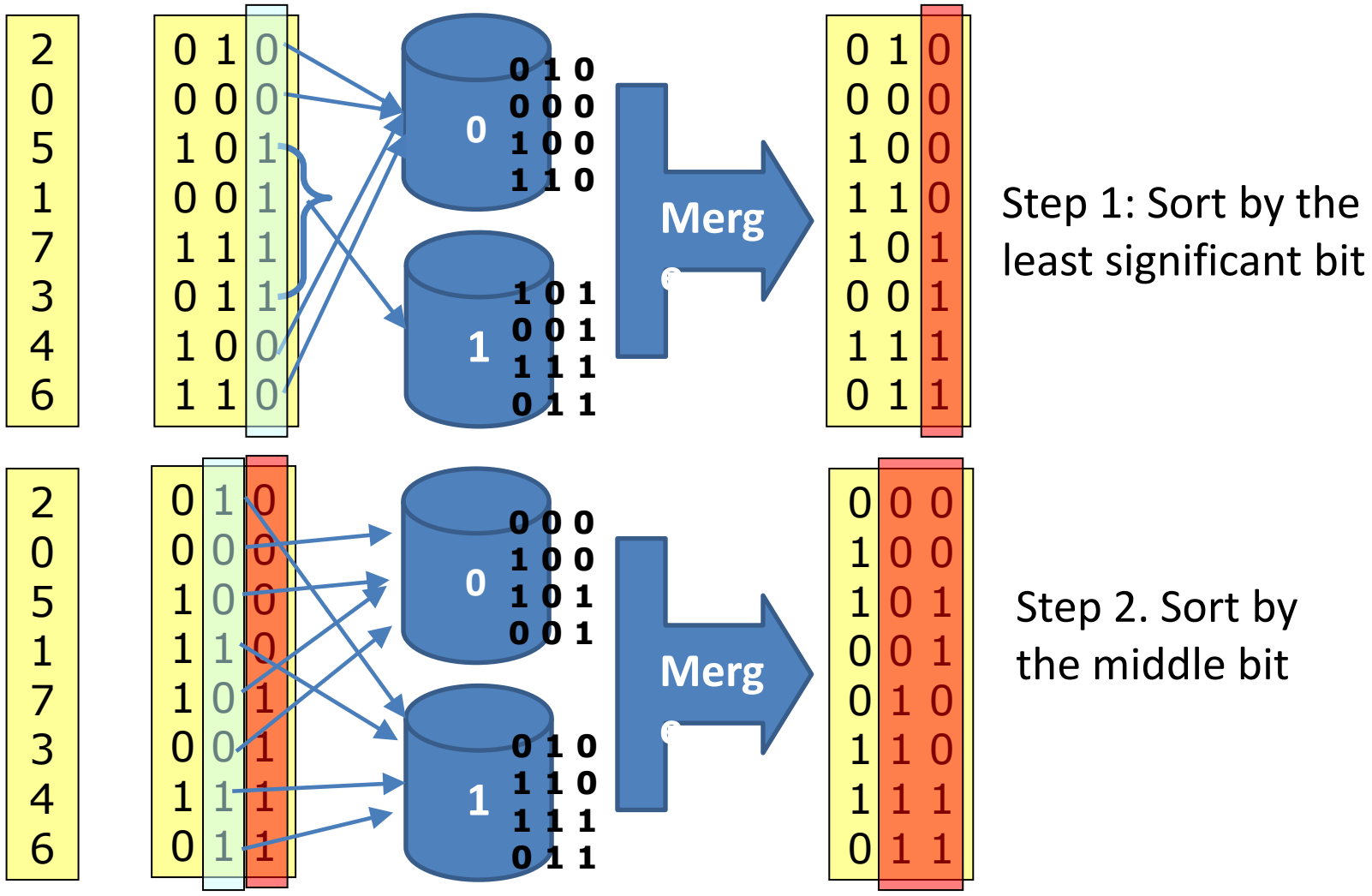
Sorted list:
0,0,0,1,1,1,2,3,3,4,4

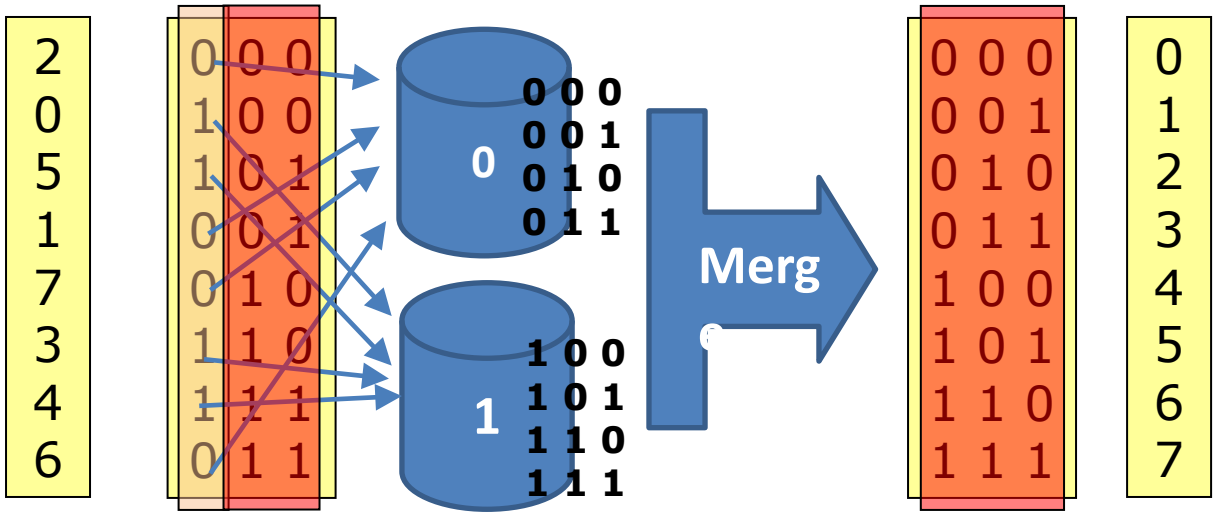
Radix Sort: bucket sort on every digit/bit

- For N elements between (L, H), using H-L+1 buckets can sort the elements in one round
- Problem: the range (L, H) may be too large.
  - Sorting 4-byte unsigned integers, range is $[0, 2^{32}-1]$ → 2^{32} buckets
- Solution(radix sort): apply bucket sort on every digit/bit

2	0 1 0
0	0 0 0
5	1 0 1
1	0 0 1
7	1 1 1
3	0 1 1
4	1 0 0
6	1 1 0

Use two
buckets 0 and 1


2 0 5 1 7 3 4 6
0 1 0
0 0 0
1 0 1
0 0 1
1 1 1
0 1 1
1 0 0
1 1 0
0
0 1 0
0 0 0
1 0 0
1 1 0
1
1 0 1
0 0 1
1 1 1
0 1 1
Merg
0 1 0
0 0 0
1 0 0
1 1 0
1 0 1
0 0 1
1 1 1
0 1 1
Step 1: Sort by the least significant bit
2 0 5 1 7 3 4 6
0 1 0
0 0 0
1 0 0
1 1 0
1 0 1
0 0 1
1 1 1
0 1 1
0
0 0 0
1 0 0
1 0 1
0 0 1
1
0 1 0
1 1 0
1 1 1
0 1 1
Merg
0 0 0
1 0 0
1 0 1
0 0 1
0 1 0
1 1 0
1 1 1
0 1 1
Step 2. Sort by the middle bit

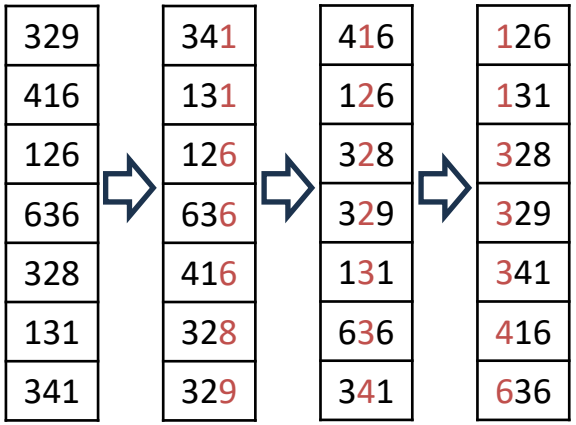


Step 3. Sort by the most significant bit

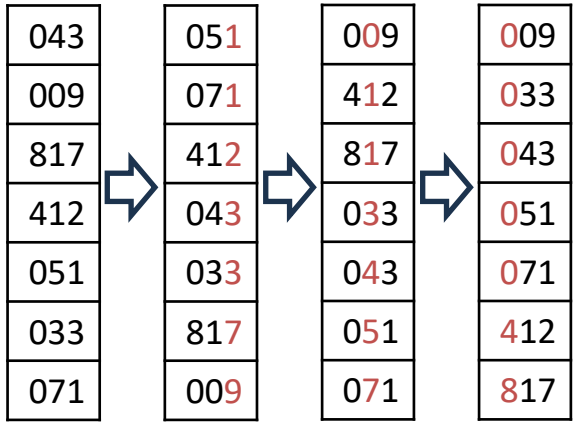
Radix Sort Animations | Data Structure | Visual How
https://www.youtube.com/watch?v=Om4BljCs_qE

You can choose an appropriate radix value

- Numbers in different formats
  - decimal whole numbers: (126, 328, 636, 341, 416, 131, 328)
  - Binary numbers: (0 001 111 110, 0 101 001 000, 1 001 111 100, 0 101 010 101, 0 110 100 000, 0 010 000 011, 0 101 001 000)
  - Octal numbers: (0176, 0510, 1174, 0525, 0640, 0203, 0510)
  - Hexadecimal numbers: (07E, 148, 27C, 1A0, 083, 148)
- Radix sort of decimal numbers using ten buckets: 0 to 9



Example 1



Example 2

References

- Radix Sort
  - https://www.geeksforgeeks.org/radix-sort/
  - https://www.geeksforgeeks.org/time-and-space-complexity-of-radix-sort-algorithm/
- Bucket Sort
  - https://www.geeksforgeeks.org/bucket-sort-2/
- Time Complexities of all Sorting Algorithms
  - https://www.geeksforgeeks.org/time-complexities-of-all-sorting-algorithms/