

Lecture 14

Radix Sort

Department of Computer Science
Hofstra University

Radix and Radix Sort

- Radix = “The base of a number system” (Webster’s dictionary)
- 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 of texts:
 - 26 if only capital letters are considered
 - 36 if capital letters and decimal digits are considered
 - 62 for capital letters + small letters + decimal digits

Radix and Radix Sort

- Radix sort was first used in 1890 U.S. census by Hollerith
- Used to sort numbers or texts
- Very efficient when sorting a large number of elements
 - $O(M*N)$. M: length of each elements; N: number of elements
- 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.

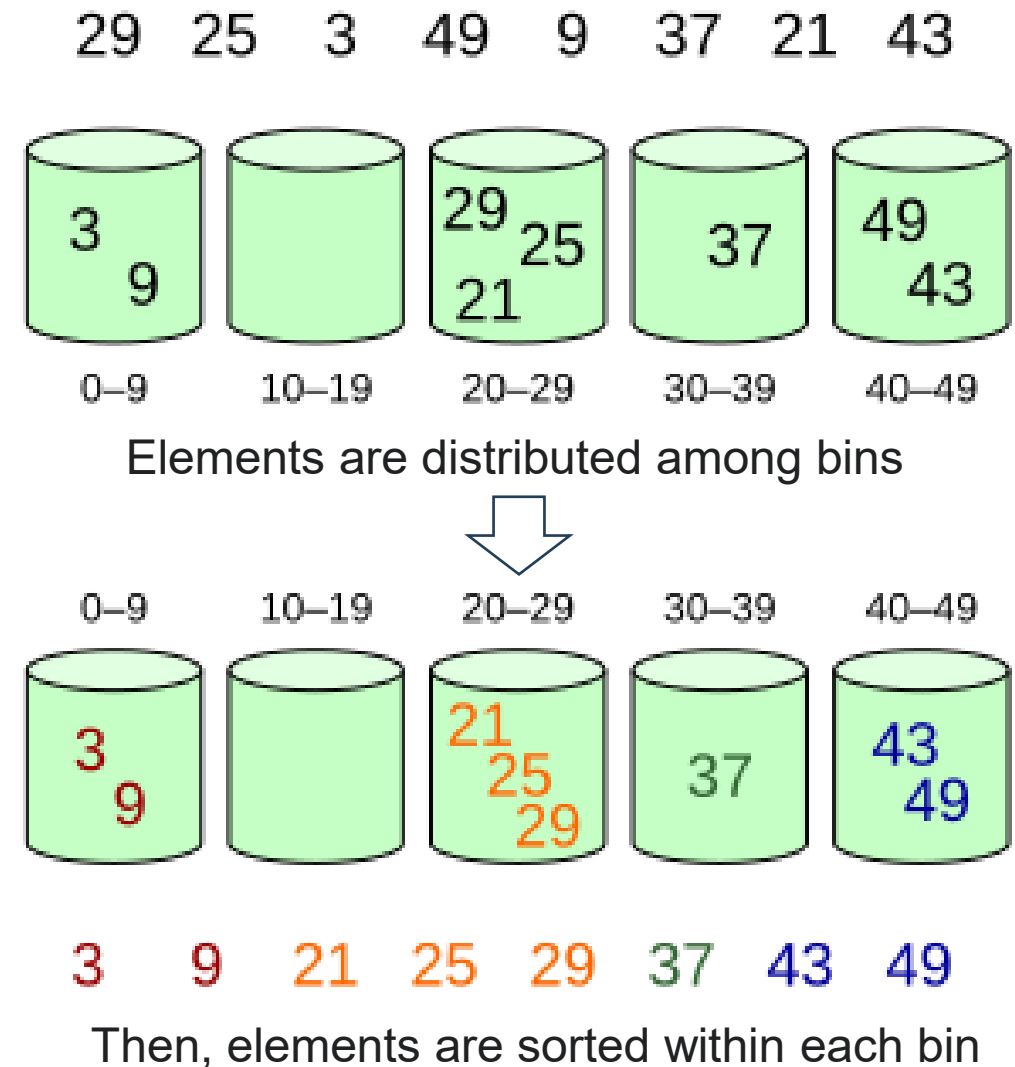
Radix Sort Algorithm

```
radix_sort(A, n, k) {  
    /* A: array; n: number of items; k: number of digits */  
    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 $O(n*k)$

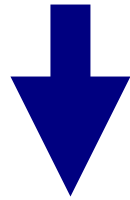
Bucket Sort

- 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.

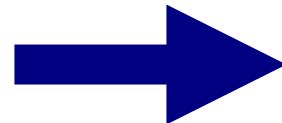


Bucket Sort as used in Radix Sort

- 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



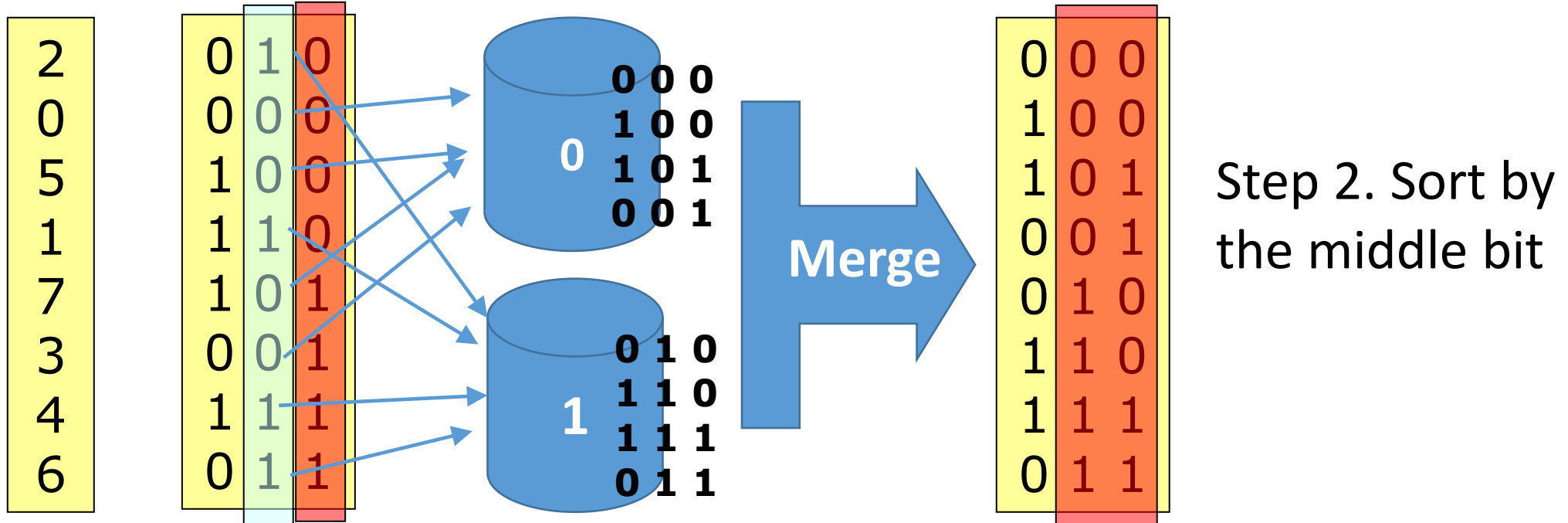
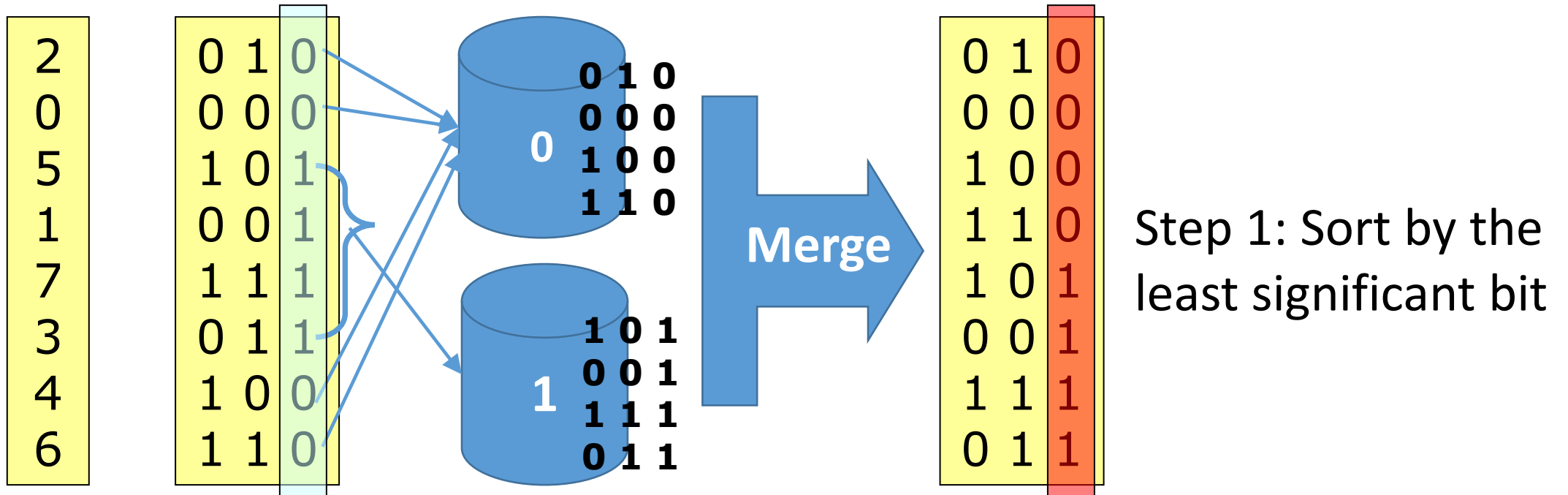
Sorted list:
0,0,0,1,1,1,2,3,3,4,4

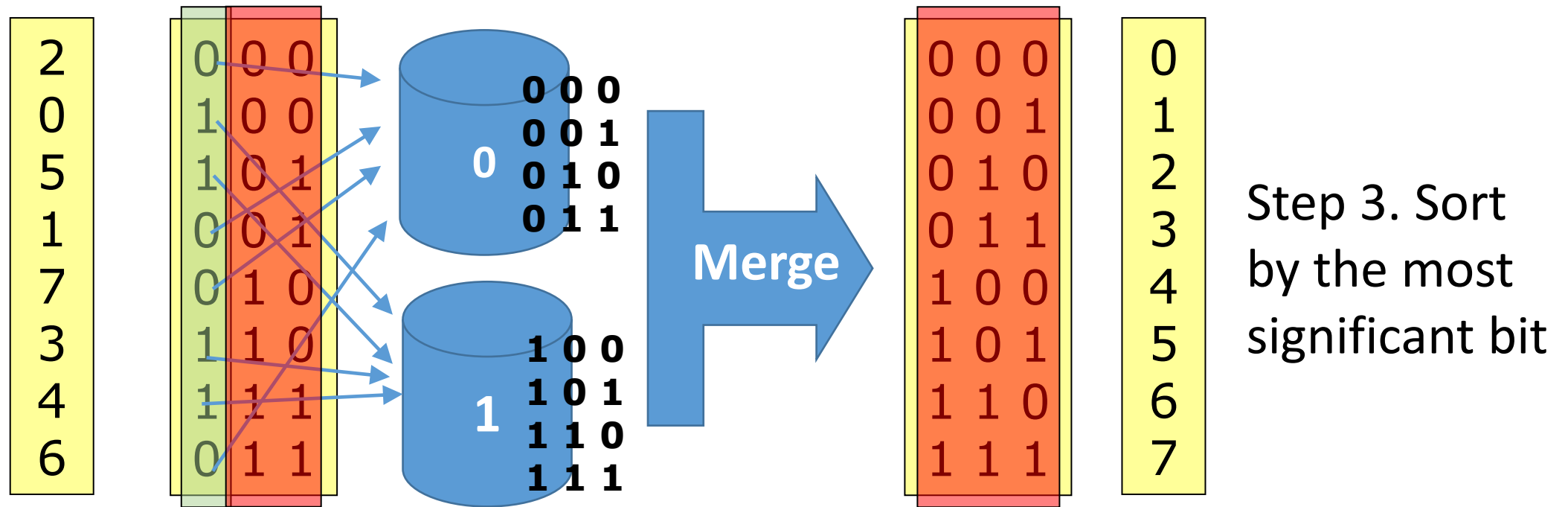
Radix Sort: bucket sort on every digit/bit

- 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 $[0, 2^{32}-1] \rightarrow 2^{32}$ 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





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

329	341	416	126
416	131	126	131
126	126	328	328
636	636	329	329
328	416	131	341
131	328	636	416
341	329	341	636

Example 1

043	051	009	009
009	071	412	033
817	412	817	043
412	043	033	051
051	033	043	071
033	817	051	412
071	009	071	817

Example 2