Design and Analysis of Algorithms
Unit -4

Bharathi R
Department of Computer Science & Engineering

DESIGN AND ANALYSIS OF ALGORITHMS

Unit 4: Space and Time Tradeoffs

Distribution Counting Sort

Slides from **Anany Levitin**

Bharathi R

Department of Computer Science & Engineering

1.Input enhancement

a)Sorting by Counting

1. **Comparison Counting Sorting**
   I. For each element of the list, count the total number of elements smaller than this element.
   II. These numbers will indicate the positions of the elements in the sorted list.

2. **Distribution Counting Sorting**
   I. Suppose the elements of the list to be sorted belong to a finite set (aka domain).
   II. Count the frequency of each element of the set in the list to be sorted.
   III. Scan the set in order of sorting and print each element of the set according to its frequency, which will be the required sorted list.

1. Input Enhancement

→ Sorting by Counting →→ ii)Distribution Counting Sorting

- A sorting method in which, with the help of some **associated information** of the elements , the elements can be placed in an array at their relative positions.

- The required information which is used to place the elements at proper positions is **accumulated sum of frequencies** which is also called as distribution in statistics.

- Hence this method is called as **Distribution counting method** for sorting.

1. Input Enhancement

→ Sorting by Counting →→ ii)Distribution Counting Sorting

Consider sorting the array whose values are known to come from the set {11, 12, 13} and should not be overwritten in the process of sorting. The frequency and distribution arrays are as follows:

13	11	12	13	12	12

Array values	11	12	13
Frequencies	1	3	2
Distribution values	1	4	6

11	12	12	12	13	13

Example of sorting by distribution counting.

13	11	12	13	12	12

Array values	11	12	13
Frequencies	1	3	2
Distribution values	1	4	6

11	12	12	12	13	13

The distribution values being decremented are shown in bold.

	$D[0..2]$			$S[0..5]$					
$A[5] = 12$	1	**4**	6				12		
$A[4] = 12$	1	**3**	6			12			
$A[3] = 13$	1	2	**6**						13
$A[2] = 12$	1	**2**	5		12				
$A[1] = 11$	**1**	1	5	11					
$A[0] = 13$	0	1	**5**					13	

Algorithm for Distribution Counting

ALGORITHM *DistributionCountingSort*$(A[0..n-1], l, u)$

//Sorts an array of integers from a limited range by distribution counting
//Input: An array $A[0..n-1]$ of integers between l and u $(l \leq u)$
//Output: Array $S[0..n-1]$ of A's elements sorted in nondecreasing order

```
for j ← 0 to u − l do D[j] ← 0                          //initialize frequencies
for i ← 0 to n − 1 do D[A[i] − l] ← D[A[i] − l] + 1     //compute frequencies
for j ← 1 to u − l do D[j] ← D[j − 1] + D[j]            //reuse for distribution
for i ← n − 1 downto 0 do
    j ← A[i] − l
    S[D[j] − 1] ← A[i]
    D[j] ← D[j] − 1
return S
```

Distribution Counting Sorting

1. This is a better time-efficiency class than that of the most efficient sorting algorithms—mergesort, quicksort, and heapsort—we have encountered.

2. It is important to remember, however, that this efficiency is obtained by exploiting the specific nature of inputs for which sorting by distribution counting works, in addition to trading space for time.

References

"Introduction to the Design and Analysis of Algorithms", Anany Levitin, Pearson Education, Delhi (Indian Version), 3rd edition, 2012. Chapter- 7

THANK YOU

Bharathi R

Department of Computer Science & Engineering

rbharathi@pes.edu