Counting Sort

Counting Sort

- Assumes that the input consists of integers in a small range 1 to k, for some integer k.
- Runs in O(n + k) time.
  - k = O(n), the sort runs in θ(n) time.
- For each element x, the algorithm
  - First determines the number of elements less than x.
  - Then directly place the element into its correct position.

Example

A[]

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

k = 6

n = 8

- Compute frequency of k elements, i.e. array C.

C[]

0	1	2	3	4	5	6
0	**2**	**0**	**2**	**3**	**0**	**1**

- Update C to store cumulative frequency.

C[]

0	1	2	3	4	5	6
0	**2**	**2**	**4**	**7**	**7**	**8**

A[]

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

C[]

0	1	2	3	4	5	6
0	2	2	4	7	7	8
0	2	2	4	6	7	8
0	1	2	4	6	7	8
0	1	2	4	5	7	8
0	1	2	3	5	7	8
0	0	2	3	5	7	8
0	0	2	3	4	7	8
0	0	2	2	4	7	7

B[]

0	1	2	3	4	5	6	7	8
							4	
		1					4	
		1				4	4	
		1		3		4	4	
	1	1		3		4	4	
	1	1		3	4	4	4	
	1	1		3	4	4	4	6
	1	1	3	3	4	4	4	6

```
for i = n – 1 to 0
    B[C[A[i]]] = A[i]
    C[A[i]] = C[A[i]] - 1
```

Algorithm

- Algorithm countingSort(A,n,k)
- Input: Array A, its size n, and the maximum integer k in the list.
- Output: The elements of A get sorted in increasing order.

```
for i = 0 to k
    C[i] = 0
for i = 0 to n – 1
    C[A[i]] = C[A[i]] + 1
for i = 1 to k
    C[i] = C[i] + C[i-1]
for i = n – 1 to 0
    B[C[A[i]]] = A[i]
    C[A[i]] = C[A[i]] - 1
for i = 0 to n – 1
    A[i] = B[i+1]
```

Thank You