DR : A Novel and Practical Noncomparison Based Sorting Algorithm

Chandresh Kumar Maurya, Neetesh Kumar* & Arun Chauhan

CSE, IIT Roorkee, *CSE, Shri Mata Vaishno Devi University Jammu
e-mail {ckm.jnu,aruntakhur,dgoldneetesh89}@gmail.com

Introduction

In the present work, we propose a novel non-comparison based sorting algorithm called DR (Divided-Remainder). Salient Features of DR are:

- It has worst case as well as expected running time complexity of $O(n\lceil\frac{\log k}{\log n}\rceil)$ where k is the largest number in the input sequence (a.k.a. *bucket size*), n is the total number of inputs.
- It does not require the uniform distribution of numbers as bucket sort does.
- It works with same time complexity no matter what is the range of input integers as required by count sort which runs in time $O(n^2)$ if the range of integers $k = O(n^2)$.
- It is stable sorting algorithm.
- It is easy to understand and implement.

DR advantages over radix sort:

- If the base of radix sort is taken as 10, DR always performs better then radix sort.
- If the base of radix sort is increased to n , then it behaves like DR in theoretical time complexity
- If the base is n, it means every digit in radix sort will use $\log_2 n$ bits, i.e. inefficient use of cache/RAM.

Working Methodology

Numbers (initial input)	index	Pass 1: dividend(Number)	Pass 2: dividend(Number)	Pass 3:
125	0	10(120),14(168)		
167	1	13(157)	1(157),1(167)	0(157),0(167),0(168),0(189)
120	2		1(168)	
115	3	9(111)	1(189)	
189	4			
111	5	10(125),7(89),10(125)		
89	6			
127	7	9(115),10(127)	0(89)	
168	8	10(128)		
128	9	15(189)	0(111),0(115)	
157	10		0(120),0(125),0(125),0(127)0(128)	
125	11	13(167)		

Running time Comparison

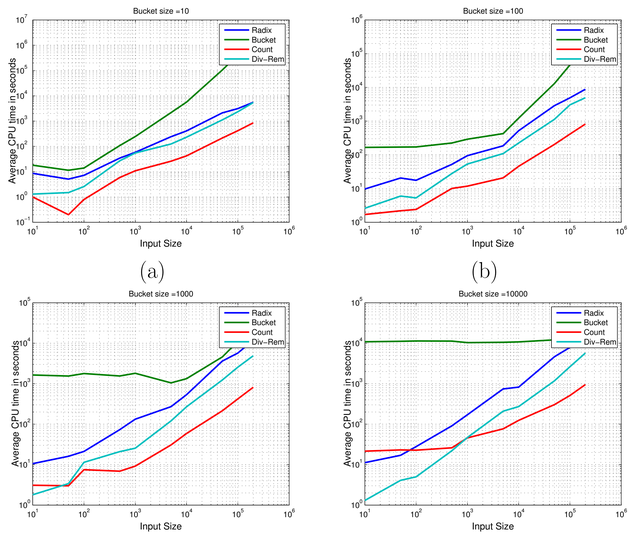

Fig. 1: Comparison of running time of radix sort, bucket sort, count sort, and DR (div-rem in the figure) sort with fixed bucket size k (a) $k = 10$ (b) $k = 100$ (c) $k = 1000$ (d) $k = 10000$

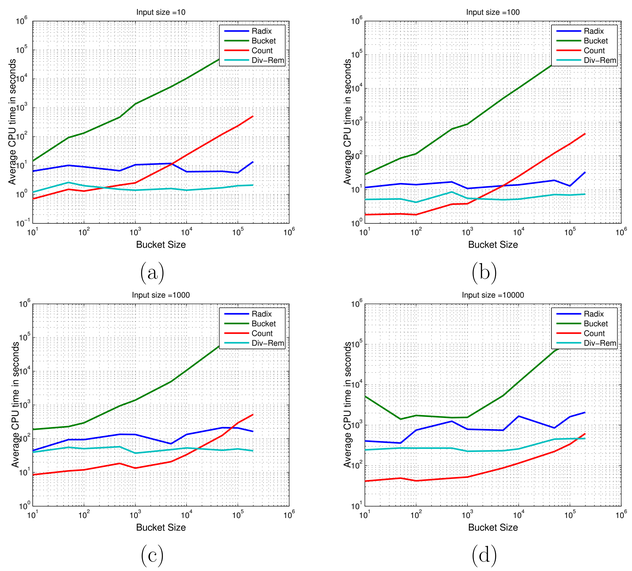

Fig. 3: Comparison of running time of radix sort, bucket sort, count sort, and DR (div-rem in the figure) sort with fixed input size n (a) $n = 10$ (b) $n = 100$ (c) $n = 1000$ (d) $n = 10000$

Conclusion

We proposed a novel and effective non-comparison based sorting algorithm referred to as Dividend Remainder (DR) sorting algorithm. The DR algorithm overcomes all the drawbacks of the count, bucket and radix sort with improved performance. The working mechanism of DR algorithm is very simple and practical as demonstrated. Further, the novelty of the DR algorithm in terms of time and space complexity (both are upper bounded by $O(n)$ (for $k < n$) is proved in the paper. Furthermore, to test the efficiency (in terms of computation time) on real systems, the DR algorithm is also implemented using two different natures of data structures: Linked list and Array. In summary, the DR algorithm is competitive to radix and count sort in terms of linked list implementation and beats Bucket sort along with effective memory managemen. It also beats three algorithms (count, bucket and radix sort) for large data values, using array implementation.