ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 1, Issue 5 , December 2014

TRIO-SORT

Goutam Mondal

Assistant Teacher, Dept of Computer Science, Belsingha Sikshayatan, West Bengal, India

ABSTRACT: In the paper, it is tried to establish a new way of sorting named TRIO- sort. The word 'Trio' means three that signifies the name. In the procedure, at first the elements are belonged in some groups where each group contains three elements (trio-elements). Taking first three elements from left, it is made first group i.e. first Trio, where second group i.e. second Trio is made by next three elements and so on until last element is belonged to last group(trio). Last group may not have three elements if the total no. of elements in the list is not multiple of three. But it does not create any problem. Sort each trio and next follow a process to insert the smallest three elements in the first trio in sorted order and so on.Although, this procedure gives the complexity o(n^2) like bubble sort and others. But it needs very few steps compare to bubble sort and any others.

I. INTRODUCTION

'sorting' means rearranging the contents in increasing order generally .Let A be a list of n elements A1,A2,A3,……,An in memory. Sorting A refers to the operation of rearranging the contents of A so that they are increasing in order either numerically or lexicographically that is, so that A1≤A2≤A3≤……≤An.

II. TRIO-SORT

Suppose an array A with n no. of elements A[1], A[2], A[3],……,A[N] is in memory. The trio-sort procedure for sorting A works as follows. First , make first trio with A[1], A[2] and A[3], then make second trio with A[4], A[5], A[6] and so on until A[N] is belonged in the last trio or group. After this, compare first and last (i.e. third) elements of each trio. If the last element is smaller than the first element of each trio, then interchange them and next, the middle element of each trio is compared with its first and last (third) elements to place it in the proper position in which the elements of each trio are sorted. After this operation, first element of the second trio is compared to the first element of first trio. If first element of the 2nd trio is smaller than the first element of first trio, then it is inserted in place of the first element of first trio and all the elements of first trio are shifted one place in the right side. Otherwise, it (first element of 2nd trio) is compared to the 2nd element of 1st trio and it is followed same operation similarly. If fail, then compare with third element of 1st trio and follow same procedure. After working with first element of 2nd trio, taking 1st element of 3rd trio same operations are performed with the three elements of 1st trio, then taking first element of 4th trio it is worked and so on. In this way, smallest three elements of the list are placed in the 1st trio in sorted order. After this, place the next three smallest elements of the list in the 2nd trio in sorted order so on. That is: Pass 1: All the trios are sorted. For this purpose, 1st and 3rd of each trio are interchanged if 1st> 3rd and then the middle is compared to existing 1st and 3rd and placed in appropriate place. After this, compare A[4] to A[1], if A[4] < A[1],then A[4] is placed in A[1] and A[1], A[2], A[3] are shifted one place in the right side. Otherwise A[4] is compared to A[2]. If A[4] <A[2], then A[4] is placed in A[2] and A[2], A[3] are shifted one place in the right side. Next if A[4] < A[3], then A[4] is placed in A[3] and A[3] is shifted one place in the right side. After this, A[7], A[10],……..till the first element of last group are compared to the elements of 1st trio and placed where necessary. Now smallest three elements of the list are placed in the first trio in sorted order.

Pass 2: All the trios starting from 2nd are sorted and taking all the 1st elements of each trio starting from A[7], same operations are performed like pass 1.

Now, next three smallest elements (i.e. 4th, 5th, 6th) of the list are placed in 2nd trio in sorted order.

……
……
……
……
……

Pass k : elements of k-th trio are sorted.

N.B. For k-th passes k =n/3, when n/3 is an integer. Otherwise k is the least integer greater than n/3.

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 1, Issue 5 , December 2014

Example: Suppose an array A contains 14 elements as follows:
66, 33, 40, 22, 55, 88, 60, 11, 80, 20, 50, 44, 77, 30

Applying the trio-sort procedure to A yields the data in the figure below. Observe that the 2nd bold and italic elements from the left is placed in place of 1st bold and italic elements from the left.

Pass	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	A[10]	A[11]	A[12]	A[13]	A[14]
	66	33	40	22	55	88	60	11	80	20	50	44	77	30
	40	33	66	22	55	88	60	11	80	20	50	44	30	77
	33	40	66	***22***	55	88	11	60	80	20	44	50	30	77
1	***22***	33	40	66	55	88	***11***	60	80	20	44	50	30	77
	11	***22***	33	40	66	55	88	60	80	***20***	44	50	30	77
	11	20	22	33	40	66	55	88	60	80	44	50	30	77
	11	20	22	33	40	66	55	88	60	50	44	80	30	77
				33	40	***66***	***55***	60	88	44	50	80	30	77
				33	40	***55***	66	60	88	***44***	50	80	30	77
2				***33***	40	44	55	66	60	88	50	80	***30***	77
				30	33	40	44	55	66	60	88	50	80	77
				30	33	40	44	*55*	66	*50*	88	60	77	80
3							44	***55***	66	***50***	60	88	77	80
							44	50	55	66	60	88	77	80
							44	50	55	66	60	88	77	80
										60	66	***88***	***77***	80
4										60	66	77	88	80
5										60	66	77	80	88
sorted	11	20	22	30	33	40	44	50	55	60	66	77	80	88

III. COMPLEXITY OF TRIO- SORT

The number f(n) of comparisons in the trio –sort procedure can be computed easily. To find the smallest elements in sorted order in the k-th pass, it requires [n-3(k-1)] comparisons where total no. of elements in the list is n. That is , there are n comparisons are required during pass 1 to find the smallest elements in sorted order them for 1st trio. And it is required (n-3) comparisons for 2nd trio and so on until m-th (where m is the least integer greater than n/3) passes for worst cases. Accordingly: $f(n)= n+(n-3)+(n-6)+\ldots\ldots\ldots\ldots\ldots\ldots+2=n(n+2)/6=O(n^2)$

IV. CONCLUSION

it is implemented a procedure in the paper that how easily a sorting problem can be solved by the trio-sort. This procedure is more efficient than the other procedure like bubble

REFERENCES

[1]. S. Lipschutz and G.A.V. Pai , "Data Structures", T. M. H. Education Pvt. Ltd.

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 1, Issue 5 , December 2014

BIOGRAPHY

GOUTAM MONDAL, received B.TECH in Computer Science from Govt. College of Engg. & Textile Technology, Serampore under West Bengal University of Technology in 2006 And M.Sc in Computer Science from Vinayaka Mission University in 2013. Now engaged as an Assistant Teacher in Belsingha Sikshayatan, W.B, India.