UVA 10810

Vishwas Karki

Problem

- The problem deals with sorting an array.
- Find the number of inversions to sort an array.
- The problem has to be solved with minimum number of inversion.
- Brute Force is not accepted.
- There are more that one way to solve the problem.

Details

Input Format

- N(first line): indicates the length of the array. (n < 500,000)
- The next N lines are integers(x).
- 0 < x < 999,999
- If N = 0 then, input is terminated

Output Format

- Print a single line of integer.
- The integer should denote the number of swaps taken to sort an array.

Solution 1: Brute force

- The number of inversions is too high and the sorting process is not efficient.
- The complexity of using brute force is O(n^2)

Solution 2: Merge Sort

- Implementing merge sort is one way to minimize the number of conversions that takes place.
- Merge sort uses recursion to disintegrate the array into single parts.
- When the elements are compared to arrange in an ascending order there is a counter that is updated whenever a swap occurs. ‘
- The complexity of using a merge sort is O(nlog(n)).