Merge sort - Lab 3

Fabio Steven Tovar Ramos

October 15th, 2018

1 Inversions

Assume that A is an array of n distinct elements.

1. Minimum number of inversions - instance.
   If A is a sorted array, the inversions will be 0.

2. Maximum number of inversions - instance.
   If A is in reverse order, the number of inversions is maximum, and will be defined by the equation 1.

$$(n-1)+(n-2)+...+3+2+1=\sum_{i=1}^{n-1} i=\frac{n(n-1)}{2} \tag{1}$$

3. Complexity (worst case number of comparisons) of the brute force counting on A.
   The complexity counting the comparisons on A by brute force is $\mathrm{O}(n^2)$, because for each element it is necessary to compare it with all those that follow it.

4. Complexity (worst case number of comparisons) of the divide an conquer (merge sort) counting on A.
   With divide an conquer technique, the complexity is $\mathrm{O}(n \log n)$, due that the counting is simplified avoiding the over-counting of elements.

2 Brute force

Run in your local machine the brute force and divide and conquer algorithms in Python 2.7 and calculate the time for the first 10^5 numbers of size instance from HackerEarth input and output and for the 10^5 sorted increasing and decreasing numbers.

Run your local machine the brute force and divide and conquer algorithms in C or C++ calculate the time for the first 10^5 numbers of size instance from HackerEarth input and output and for the 10^5 sorted increasing and decreasing numbers.

Divide and Conquer.
HackerEarth input: 0.285416 seconds
Increasing order: 0.018375 seconds
Decreasing order: 0.018182 seconds
Brute force.
HackerEarth input: 0.285416 seconds
Increasing order: 0.018375 seconds
Decreasing order: 0.018182 seconds