Тураев Тимур, 504 (SE)
20.09.2013

3.3 Дан массив из n различных элементов. Требуется найти число инверсий за O(n log n).

Задачу решает модифицированный алгоритм сортировки слиянием. В момент слияния смотрим, если происходит вставка в окончательный массив элемента из второго массива (то есть в первом, левом, массиве элемент оказался больше чем в правом – инверсия), то дополняем счетчик инверсий на n1-i, где n1 – длина левого (первого) массива, а i – индекс текущего элемента в этом массиве. Данная разность есть ни что иное как длина непросмотренной части – то есть именно столько инверсий будет порождать наш элемент во втором массиве (надо помнить о том, что массивы уже отсортированы по возрастанию) – так как все последующие элементы в первом после i-того будут больше, чем текущий элемент из второго массива – то есть все они породят инверсии.

Осталось понять, почему при сортировке частей не будет потеряно и приобретено ни одной инверсии. Но это очевидно: предположим, что какие то два элемента образуют инверсию. Тогда посчитаны они могут быть лишь в одном случае – когда они оба одновременно окажутся в разных массивах-параметрах, передаваемых методу Merge. Но несложно сообразить, что такая передача будет лишь один раз – во-первых когда-нибудь такая передача состоится, ибо мы делим массив на 2, пока они не станут размером 1, во-вторых после передачи этот массив «сливается» и больше никогда не уменьшается. Аналогичные рассуждения можно провести и с элементами, которые инверсию не вызывают.

Итоговое количество инверсий:

```
res = 0;
res = mergesort(a, l, q) + mergesort(a, q+1, r);
res += merge(a, l, q, r);
```

Сколько было в левой половине + сколько было справа + сколько появится при сливании массивов.

Сложность – очевидно не асимптотически выше, чем у сортировки слиянием: O(n log n)

Код (работающий =)): http://pastebin.com/fL8Lzc6d