

CKA::sort()

```
template <class RandomAccessIterator, class Less>
void sort(RandomAccessIterator first, RandomAccessIterator beyond,
Less less)
```

```
template <class InputIterator, class OutputIterator, class Less>
OutputIterator sort(InputIterator first, InputIterator beyond,
OutputIterator result, Less less)
```

sorts elements in the range [**first**, **beyond**) with a comparator **less**

It uses **merge sort** to sort elements, but isn't completely non adaptive. It checks if partition of array sorted is in normal and reversed way.

Parameters

- **first**, **beyond**: the range of elements to sort
- **less**: comparison function object
- **result**: the first element of output

Returns

- a OutputIterator of output's last element