

## 5.1 Merge sort template

---

**Due** 7 Oct 2019 by 23:59      **Points** 0      **Submitting** an external tool

---

For this assignment you will implement the *merge sort* algorithm. *Merge sort* is a recursive algorithm from the class of divide-and-conquer algorithms. For details how it works, please see the syllabus of *Computational Thinking*. *Merge sort* does not sort "*in place*", meaning it does not sort the vector of input values but uses another vector to store the sorted values. Actually, *merge sort* could be implemented to sort "*in place*", but it is difficult to get it right, so we do not aim at it. Side note: it is perfectly OK if your *merge* phase actually makes another copy of the values you are merging.

To give this assignment another twist, we want you to write *merge sort* as a template function that works independently of the type of elements to be sorted. For this, we want you to write a file **mergesort.h** that implements a template function according to the following declaration:

```
template <typename T> void mergeSort(const std::vector<T>& unsorted,
                                     unsigned int firstToSort,
                                     unsigned int lastToSort,
                                     std::vector<T>& sorted);
```

*mergeSort* takes a vector of *unsorted* values. An invocation of *mergeSort* sorts the elements from *unsorted.at(firstToSort)* up to *unsorted.at(lastToSort)*, and stores them in sorted order in the *sorted* vector, at the same positions, *firstToSort* until *lastToSort*. You can safely assume that the vector *sorted* has exactly the same size as *unsorted*.

For testing your template function, you can use the following two main programs that are identical to each other, except for the type of vector elements to be sorted:

- [mergesortdouble.cpp](#) 
- [mergesortstring.cpp](#) 

You may need to use the option **-std=c++14** when compiling these programs with *g++*.

This tool needs to be loaded in a new browser window

Load 5.1 Merge sort template in a new window

