

# Advanced Programming Lab Exercise 02

## 1 Optimizations

### 1.1 Count Sort

#### 1.1.1 Incrementation

Incrementation or decrementation in the style `var++` was replaced by `++var`. This way the value is returned before incrementation and thus less memory is used.

This change didn't have any significant effect on execution time. Probably because returned variables are not used anywhere in the code and so compiler didn't include them in the first place. The change was left in a final code as it is a better coding practice.

#### 1.1.2 Loop Unrolling

Middle loop in the function was unrolled. We consider two cases: If the `range` is even, we can unroll the loop and do "two iterations in one". If it's odd we make first step and then unroll rest of the loop.

Unrolling first and third loop didn't have any effect on the execution time. It's probably because this has already been done by the compiler as the loops are simple.

### 1.2 Radix Sort

#### 1.2.1 Incrementation

Same results as for Count Sort.

#### 1.2.2 Loop Unrolling

We have experimented with loop unrolling but it had no or even negative effect on execution time. It's probably because this has already been done by the compiler as the loops are simple. Negative effects might have been caused by the fact that in order to unroll the loop we needed to use integer and obtain size of the array instead of more efficient loop with iterator e.g. `for(auto &e:vector)`.

#### 1.2.3 Changing to binary sorted

16 buckets were used instead of 10 and therefore dividing by 10 was replaced by efficient dividing by 10 which can be implemented as bit shift `<< 4`.

This was the only change which significantly improved execution time and was kept in the final code.

Average time in ms from 10 000 tests for 10 000 elements in range [0..65535]:

	count sort	radix sort vector	radix sort list
unoptimised	0.252521	0.312107	0.445378
optimised	0.15476	0.146311	0.253495

## 2 STL algorithm comparison

Average time in ms from 10 000 tests:

range	elements	count sort	bin. radix vector	bin. radix list	STL vector	STL list
255	1000	0.00394284	0.0143949	0.0185247	0.0230164	0.0671095
255	10000	0.0277105	0.0946977	0.135941	0.407291	1.0628
255	1000000	0.423554	0.885112	1.47969	3.9535	21.2646
65535	1000	0.0902461	0.0265267	0.0327888	0.0222939	0.067683
65535	10000	0.158156	0.171707	0.259274	0.4993	1.09692
65535	1000000	1.57418	1.68161	2.85029	6.23986	19.3583