



Efficient Programs

Group 15

Ellhotka Yannic
Wesp Simon
Palikarsky Konstantin Konstantinov
Singh Abhinandan
Basyrov Ruslan
Gutak Danil
Pfisterer Daniel



Implementation

C

Read CSV

Sort by column

Join on column

Output data





Optimizations

Base Performance: 219 billion cycles



inlining

Copy code instead of calling a function multiple times

Better performance

Worse binary size

218 billion cycles



Custom quicksort

Replace stdlib qsort with custom implementation

213 billion cycles



mmap

Map file directly into memory

Avoid explicit read() or write() system calls

Less user <-> kernel space operations

175 billion cycles



replace strndup

With manual malloc and memcpy

Avoid unnecessary string length comparisons

173 billion cycles



replace snprintf and strdup

Use pointers and dynamically allocated buffers instead

168 billion cycles



Buffered output

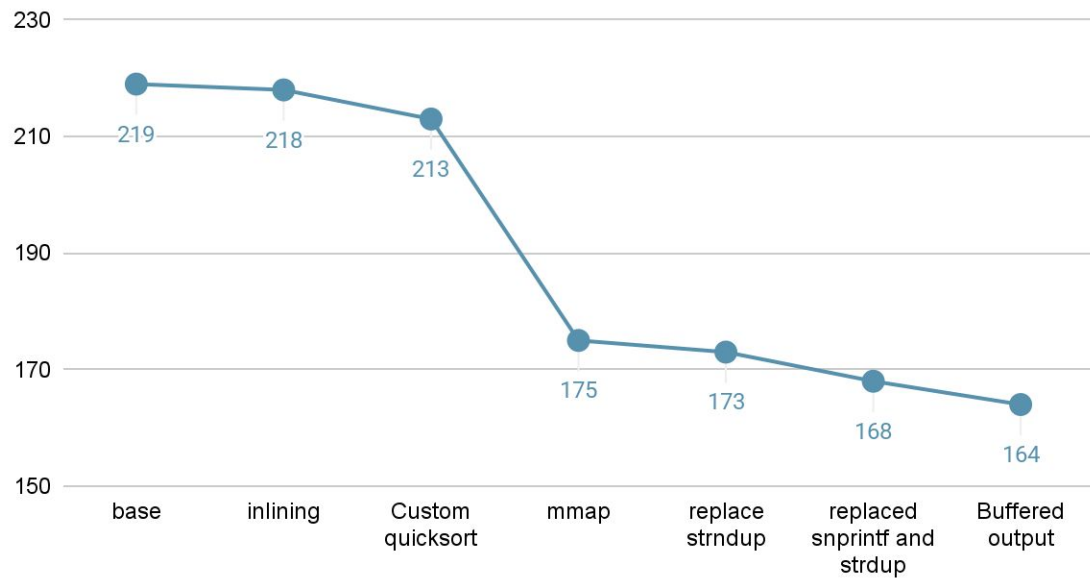
Instead of printing every line separately

Print one big string

164 billion cycles



Billion Cycles





Summary

Allocating memory by hand is usually faster

Stdlib functions may take extra steps for convenience -> bad for performance

Repository:

<https://github.com/smonist/2024WS-efficient-programs>