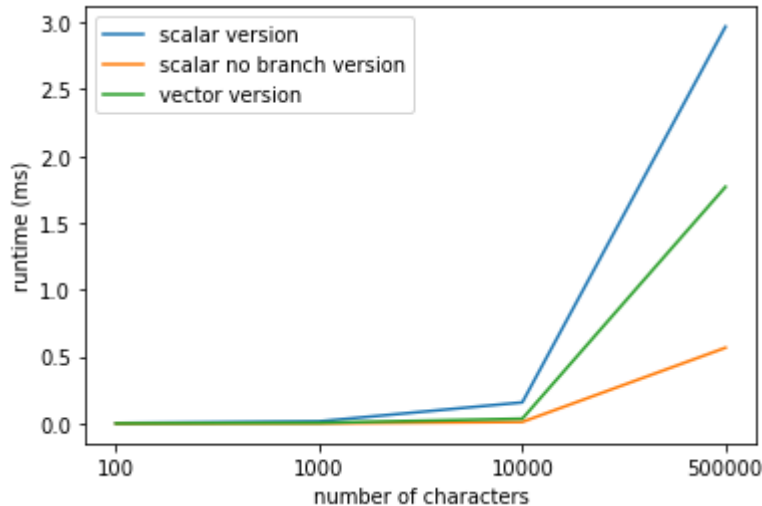


Report - SIMD Vectorization

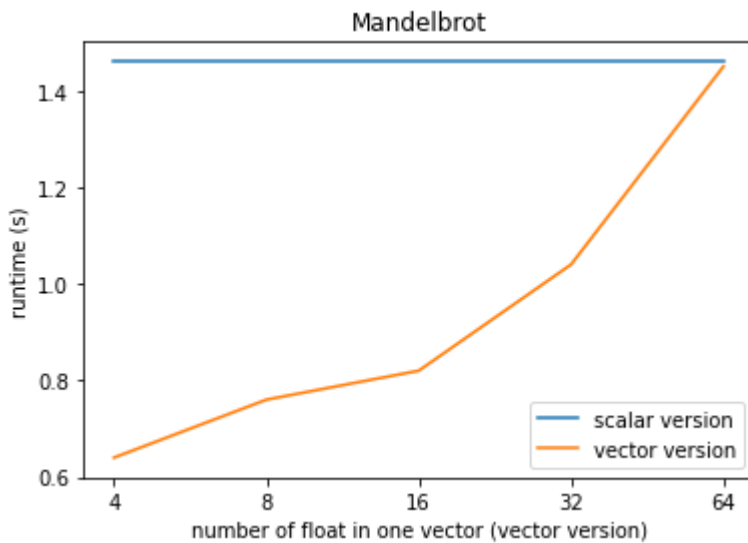
Characters to lowercase:

- For x86-64 gcc 13.1, there is no auto-vectorization of the scalar implementation using all optimization settings (O1,O2,O3). For x86-64 clang 15.0.0, there is no auto-vectorization using -O1, but it will auto-vectorize the code under -O2 and -O3.



- From the figure, we can see that both vector version and scalar no branch version are faster than the normal scalar version implementation. And the scalar no branch version is the best one. The no branch version is very efficient. I think the reason no branch is quicker than vector version, is that there is no extra work (compared with the vector version) for scalar no branch version.
- When the input characters size is large enough, vector version can be twice as quick as the normal scalar version.

Mandelbrot:



- vector version Mandelbrot with small **vector_size** is much faster than the normal scalar version.
- When I try to use largest possible size of vectorization, it becomes slower. Because the Mandelbrot is inherently sequential in nature due to the dependence on the previous iteration's results. And with larger **vector_size**, a lot of time will be spent on the loop to do some extra work.