

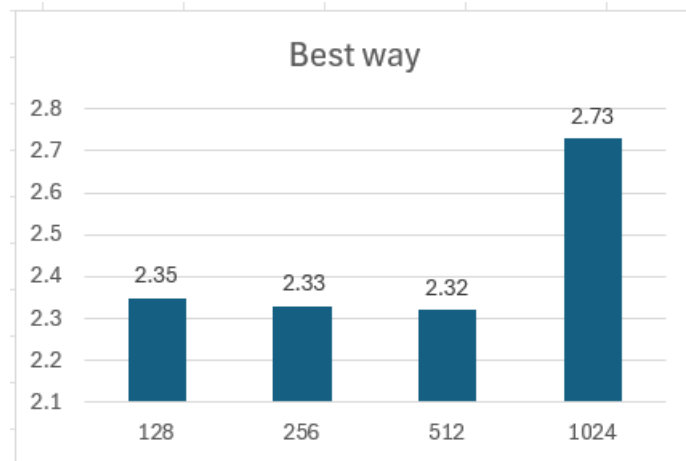
# Assignment 2: Efficient Program Implementation

202011133 Hyeonsoo yun

2024.06.16.

## 1. Best way

The vertical axis of the graph is speedup, and the horizontal axis is the number of pixels on one side of the input image. Speedup was measured 5 times and the average value was used.



1. Code motion: The operation within the for statement was taken out of the for statement.
2. Sharing of common: Reuse rate was increased by putting repetitive calculations in multiple lines into new local variables.
3. Elimination of dynamic memory allocation: Dynamic memory allocation was reduced to make compiler optimization easier.
4. Use of restrict keyword: The restrict limiter was used to prevent the compiler from optimizing properly due to potential memory aliasing issues. By using the restrict keyword to tell the compiler that pointers do not overlap each other, the compiler can perform more aggressive optimization.
5. Boundary condition check optimization: Redundant boundary condition checks were minimized and processed efficiently within the loop. For example, unnecessary testing was reduced by processing multiple boundary conditions at once depending on the range of x and y coordinates.
6. Minimize index calculation: Index calculation is performed in advance to reduce the number of operations.
7. Instead of calling the function, the convolution function was inserted into the filter\_optimized function.

## 2. different approach

- try 1: By processing two pixels simultaneously (loop unrolling), I created an environment where compilation is easy to optimize.
- try 2: By processing three pixels simultaneously (loop unrolling), I created an environment where compilation is easy to optimize. Since it is 3 times 3 filters, I thought unrolling 3 at a time would have

better results, but the speedup was actually reduced compared to unrolling 2 at a time. Therefore, the optimal unrolling coefficient was determined to be 2.

-try 3: Instead of putting the convolution function inside the filter\_optimized function, copy and paste it using inline. We also use the const modifier to indicate that the variable will no longer change.

-try 4: I added loop unrolling, which calculates two operations at a time, to the best way method, but the speedup actually decreased.

-try 5: After finding out the size of the Raspberry Pi's cache, I set a block that matched the cache size and tried to change the number of loops by the block size. However, I also tried 'lscpu', 'getconf LEVEL1\_DCACHE\_LINESIZE' and 'sudo dmidecode -t cache' but could not find the cache size of the Raspberry Pi.

