Homework #5: Optimization

November 20th, 2019

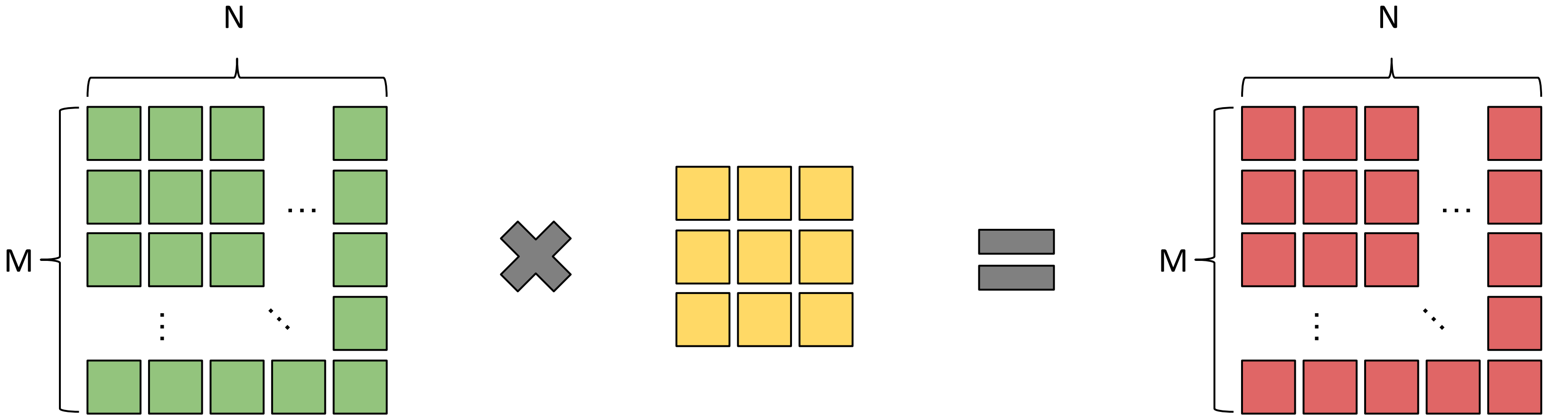
Jae W. Lee (jaewlee@snu.ac.kr)
Computer Science and Engineering
Seoul National University

TA: Jeonghun Gong, Yunho Jin

Goal of This Project

- **Optimize the given matrix convolution operation.**

Convolution (Simplified)



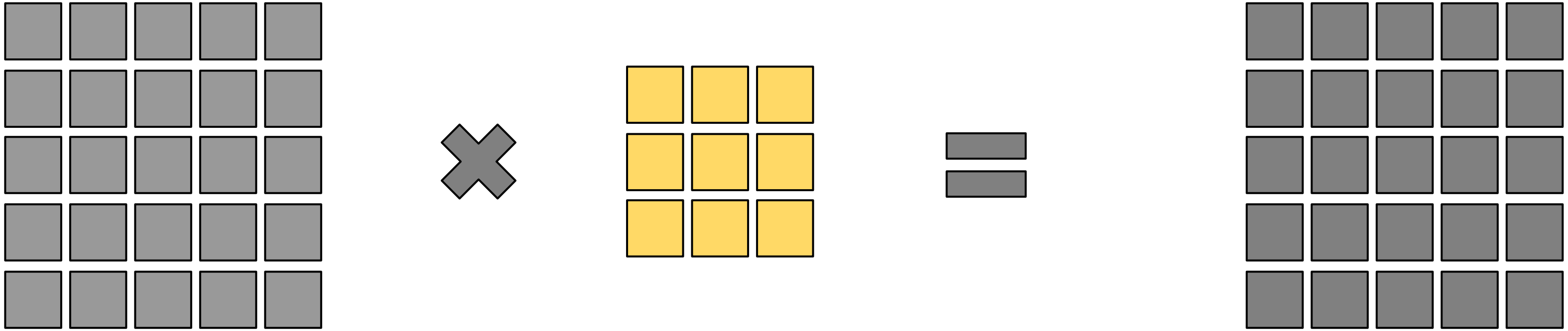
Input image (M x N)

Filter (3 x 3)

Output image (M x N)

*Assume stride = 1

Convolution Example

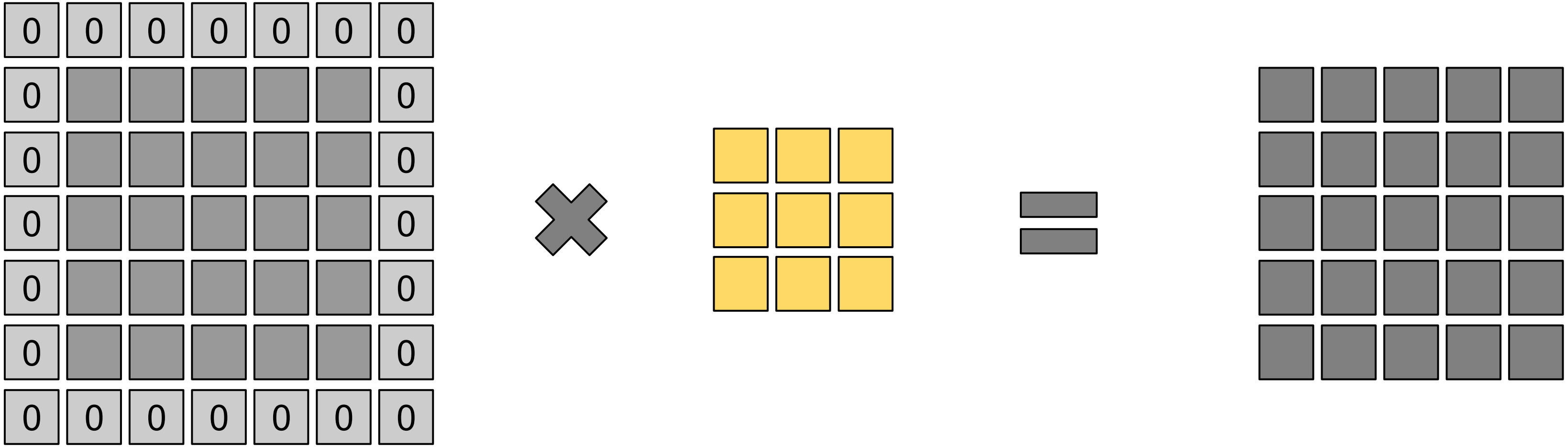


Input image (5 x 5)

Filter (3 x 3)

Output image (5 x 5)

Convolution Example



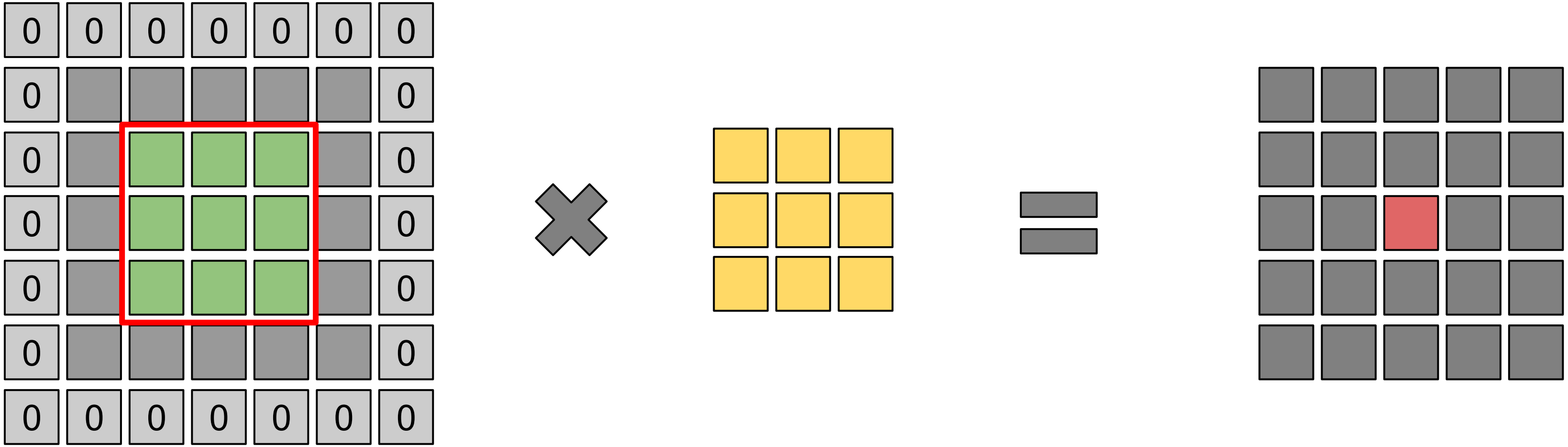
0	0	0	0	0	0	0
0						0
0						0
0						0
0						0
0						0
0	0	0	0	0	0	0

Padded input image (5 x 5)

Filter (3 x 3)

Output image (5 x 5)

Convolution Example



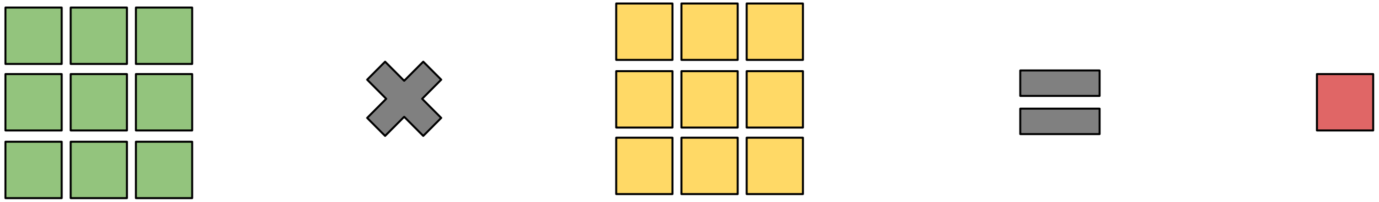
Padded input image (5 x 5)

Filter (3 x 3)

Output image (5 x 5)

Convolution Example

```
#For each output pixel
O = 0;
for(i = 0 ; i < 3 ; i++)
    for(j = 0 ; j < 3 ; j++)
        O += I[i][j] * filter[i][j];
```



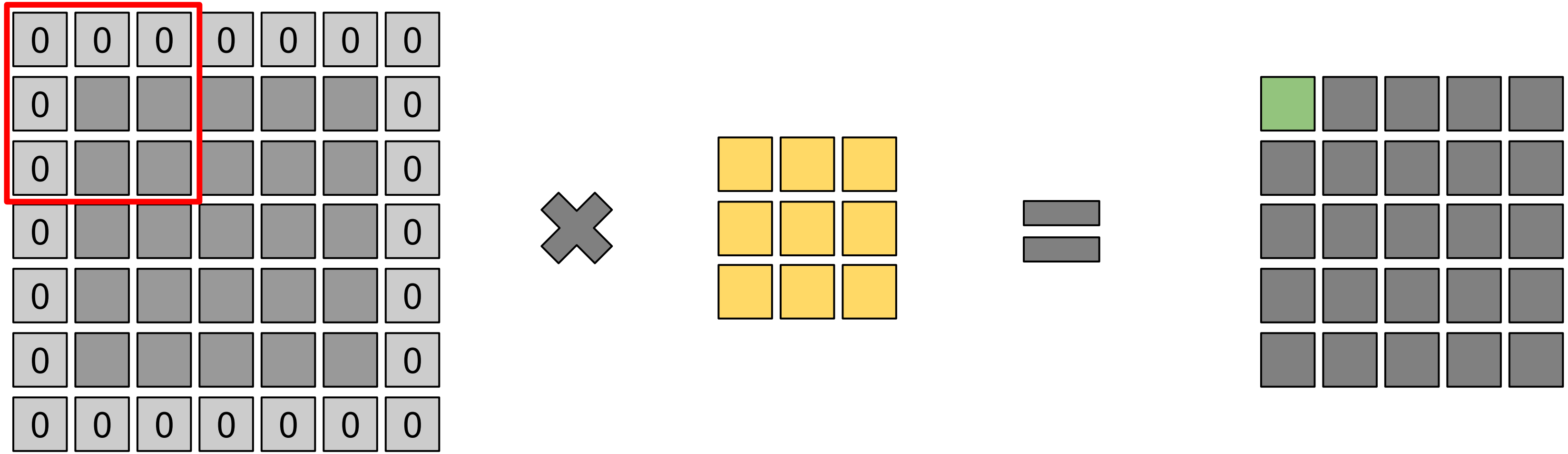
Part of input image(I) (3 x 3)

Filter (3 x 3)

Output Pixel(O)

Convolution Example

- **Repeat it for every output pixels**



0	0	0	0	0	0	0
0						0
0						0
0						0
0						0
0						0
0	0	0	0	0	0	0

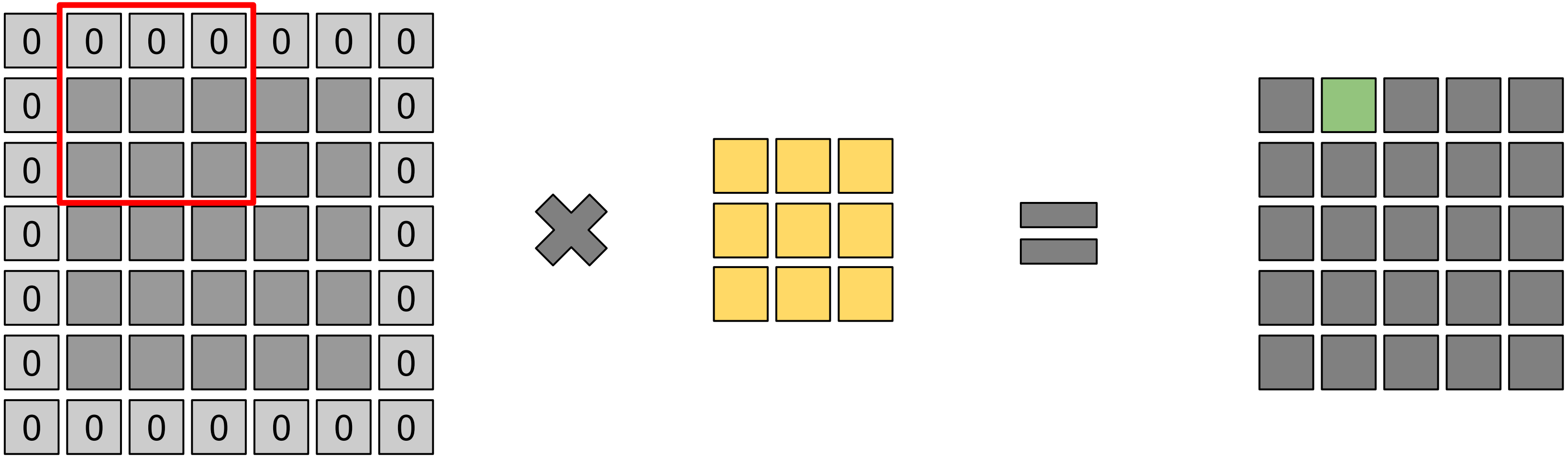
Padded input image (5 x 5)

Filter (3 x 3)

Output image (5 x 5)

Convolution Example

- **Repeat it for every output pixels**



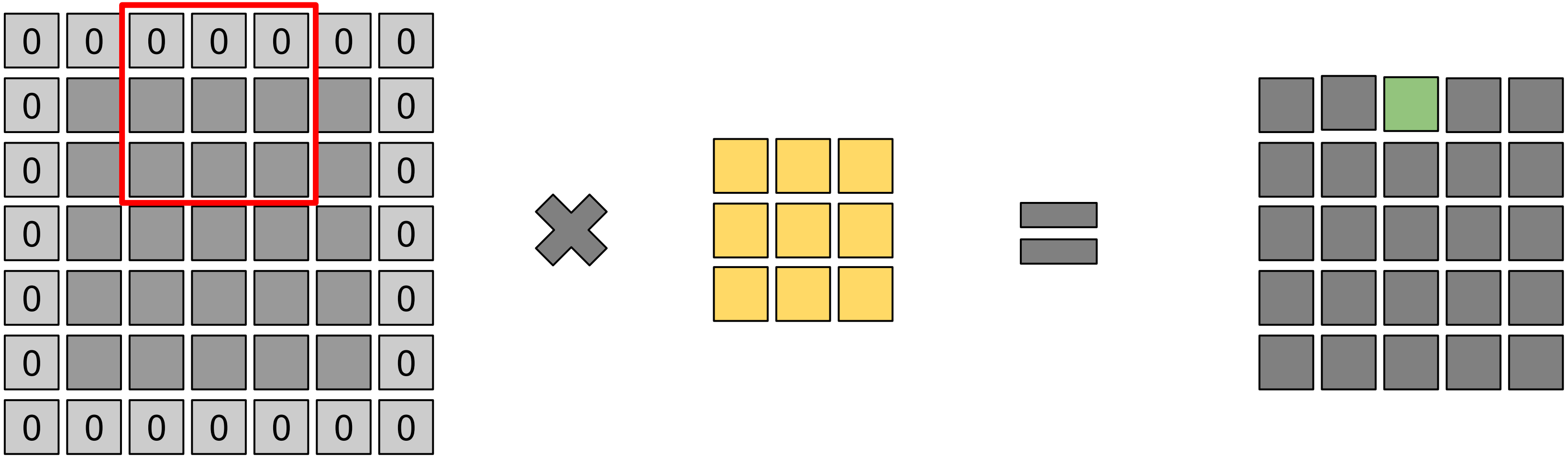
Padded input image (5 x 5)

Filter (3 x 3)

Output image (5 x 5)

Convolution Example

- **Repeat it for every output pixels**



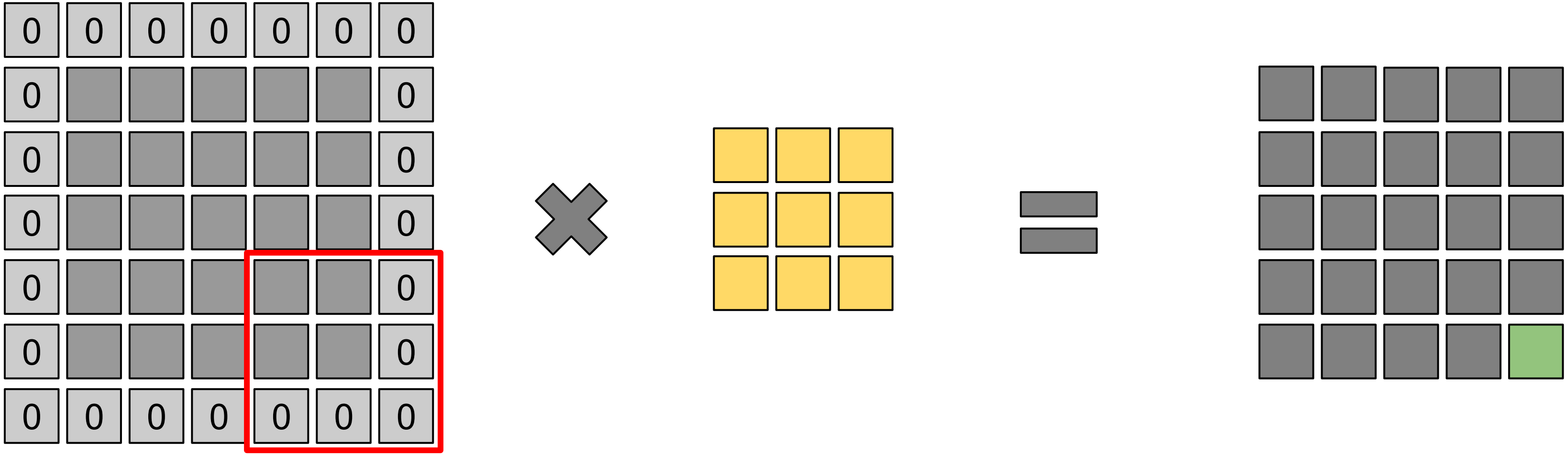
Padded input image (5 x 5)

Filter (3 x 3)

Output image (5 x 5)

Convolution Example

- **Repeat it for every output pixels**



Padded input image (5 x 5)

Filter (3 x 3)

Output image (5 x 5)

What You Need to Do

- **Optimize the given convolution with...**
  - Blocking
  - Loop unrolling
  - Reduce branch penalty
  - SIMD operations
  - Inline assembly (you need to study x86_64 ISA)
  - Many others...
  - But, **DO NOT** write multi-threaded program or use accelerators (e.g., GPUs)

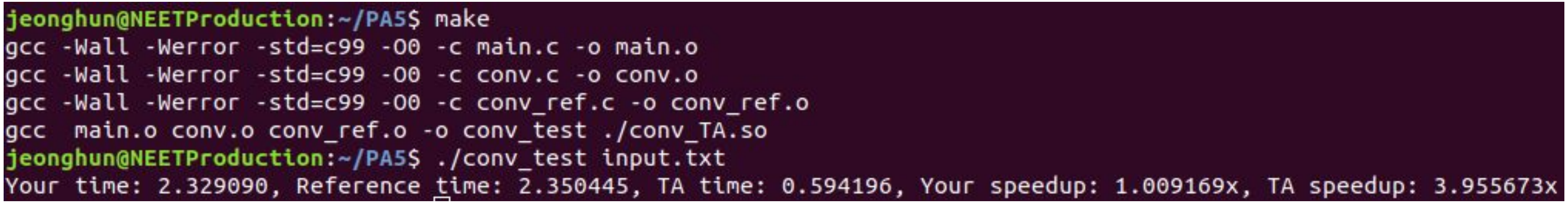
- **File to modify (and submit):** `conv.c`

Setup

- **You may use given VM instance or HW lab computers for development**
  - Performance will be measured using HW lab computers
  - We strongly recommend you do measurement at the HW lab before submission.
- **Download PA5.tar from eTL**
  - Extract it on your directory
  - On terminal:
    ```
    $> tar -xzvf PA5.tar.gz
    ```
  - Build:
    ```
    $> make
    ```
  - Execute your code:
    ```
    $> ./conv_test [input file]
    ```

Grading Rules (1)

- **We will grade your submitted code on a HW lab computer**
  - We will measure the performance of your code using the same input.txt.
  - All execution times will be measured by the minimum of 5 runs.
  - If you do hardware-specific optimization (e.g., cache optimization, SIMD), you must target the HW lab computer.



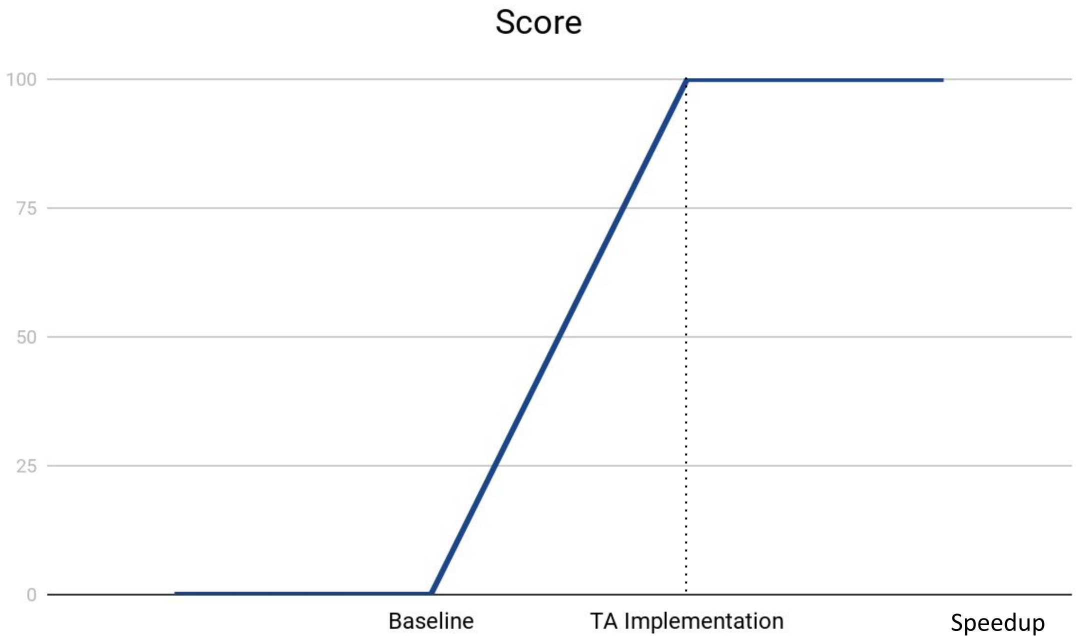
```
jeonghun@NEETProduction:~/PA5$ make
gcc -Wall -Werror -std=c99 -O0 -c main.c -o main.o
gcc -Wall -Werror -std=c99 -O0 -c conv.c -o conv.o
gcc -Wall -Werror -std=c99 -O0 -c conv_ref.c -o conv_ref.o
gcc  main.o conv.o conv_ref.o -o conv_test ./conv_TA.so
jeonghun@NEETProduction:~/PA5$ ./conv_test input.txt
Your time: 2.329090, Reference time: 2.350445, TA time: 0.594196, Your speedup: 1.009169x, TA speedup: 3.955673x
```

Grading Rules (2)

- **Other rules**
  - If your code is faster than TA’s, you will get 100 points.
    - If it is slower, we will use a grading curve on next slide.
    - Top *k* students will be given bonus credits by up to 100 points.
      - *k* is around 3-5 depending on the outcome.
      - We’ll assign bonus points according to the speedup numbers.
  - We will use gcc on Linux with a fixed set of flags
    - Refer to attached Makefile.
    - Playing with compilation flags is not allowed.
  - If your code is incorrect, you will get a very low score.
    - We may test the correctness of your code using other inputs.

Grading Rules (3)

- **Grading curve**



Submission guideline

- **Write-up**
  - Briefly describe your implementation
  - File name should be [Student ID].pdf (example: 2019-12345.pdf)
- **Compress your code (conv.c) and write-up in a single ZIP file.**
  - File name should be [Student ID].zip (example: 2019-12345.zip)
- **Due: 2019.12.9 (Mon) 23:59 KST**
  - Within next 24 hours: -10% deduction
  - Within next 48 hours: -30% deduction
  - Within next 72 hours: -50% deduction
  - After next 72 hours: Submission not accepted