



---

## GOAL

Apply the code transformations discussed in lecture (along with a couple of others) to lower the execution time of four different algorithms:

1. matrix initialization,
2. array initialization,
3. factorial computation, and
4. matrix multiplication.

You will modify the starter code `copt.c` and measure your code transformations using `test.sh`.

---

## PROCEDURE

First, download the `copt.tar.gz` archive and expand it in your Linux environment using `$ tar xzvf copt.tar.gz`.

You will only write code in the four “optimized” functions within the application (and, potentially, “optimized” helper functions). You should not modify any other code in `copt.c`. You should consider a testing strategy: implement one transformation at a time, then measure the effect of the transformation after compilation (you might want to further modify the included `Makefile` to help speed up this process).

Document the transformations you make to a function in a block comment directly above the “optimized” function signature.

---

## CODE TRANSFORMATIONS

Here is a list of some of the code transformations you may want to apply to your starter code:

1. Code motion,
2. Strength reduction,
3. Function inlining,
4. Common subexpression elimination,
5. Loop unrolling,
6. Tail recursion elimination,
7. SIMD operations (AVX instructions), and
8. Suggest variables to the compiler for mapping to registers.

There may be other code transformations you can apply to further speed up execution.

---

## DELIVERABLES

Submit your `copt.c` source code with transformed functions, (optional) `Makefile`, and the final (or best) output from the `test.sh` script (as a `.txt` file) to the Dropbox by the posted due date.