KOÇ UNIVERSITY

COMP 201 - Fall 2021
Lab Exercise 9: Code Optimization
Due: 10 Jan 2022 23:59

This lab covers some basic techniques of code optimization. The code optimization in the synthesis phase is a program transformation technique, which tries to improve the intermediate code by making it consume fewer resources (i.e. CPU, Memory) so that faster-running machine code will result.

In this exercise, you are given a base code with some implemented functions. However, the implementations are simple and not optimized, and your aim is to optimize these functions as the manual requested. You can compile the source code with ***`make`*** command (**makefile** is provided). Further, you can call different functions to get tested in terms of speed of execution.

1. To test Matrix multiplication with input size **`2000 × 2000`** run:

   `make matmul`

2. For testing foo1 function with input size **`10000 × 10000`** run:

   `make foo1`

Note that you can change input sizes on the makefile if you are facing a problem with low speed during the debugging phase. But remember, you need to report final results with the given input sizes.

You are asked to perform the followings:

1. Optimize ***`opt matmul`*** function with 10 times loop unrolling.
2. Report the execution time results and corresponding speedup.
3. Other than loop unrolling, if you can find any other optimization (other than compiler optimization), it is a plus point, and you should add it to your report.
4. Optimize ***`opt foo1`*** function with in-lining technique and report the results.
5. Improve the for loops in ***`opt foo1`*** function (other than loop unrolling) and report the results.
6. Finally, re-run the experiments but this time use compiler optimization level 3

   (**Hint**: check the makefile!) and report all previous results and notice how simple implementation performs.

1 Submission

Submit your **main.c** code file and a simple report in **report.pdf** format that shows experiments and their time and speedups. You can mention any extra point you found or performed in the report. It is a nice experience to increase input size from a small number and see how speedup is changing.