

Homework 3 - Sparse Matrix Vector Multily (SpMV)

Jee Whan Choi & Chris Misa

April 26, 2019

DUE DATE: 11:59 PM 5/3/2019

The objective of this homework is to implement different data structures for a sparse matrix and use them to calculate sparse matrix-vector multiply. The homework has three parts

- Given a CSR format for a sparse matrix, calculate the SpMV.
- Given a CSR format, conver it to a COO format.
- Using the COO format you constructed, calculate the SpMV using COO.

Follow the instructions below to get started:

- In your class repo, create the directory structure `homeworks/hw03/`. The rest of this assignment should be completed in the `hw03` directory which is where the grader will look for your final solution.
- Once the tarball has been extracted, you should see six files: this pdf, `hw03_spmv_skeleton.c`, and four test files (*.dat).
- Compile the code using `gcc hw03_spmv_skeleton.c` which will generate the `a.out` executable.
- Study the code to see what is happening. This is **important** since you will use some or all of these functions to implement the homework.
- As before, rename your file to `hw03_spmv_answer.c` before committing it to the repository.

hw03_spmv_skeleton.c This program takes no input from the user. The main function calls a sequence of functions - *load_matrix*, *load_vector*, *print_sparse_matrix*, etc. Go through them to see how the matrix and vector are stored.

- The four test files work in pairs - `sparse_matrix.dat + vector.dat` is one pair, and `sparse_matrix.1.dat + vector.1.dat` are another, and the sparse matrix will be multiplied to the corresponding vector. These are provided so that you can test your code on different sizes inputs. Change the `#define` at the top to use different files as input.
- *res_dense* contains the result of the SpMV between the input sparse matrix (`int_array`) and input vector (`int_vector`). This is the ground truth against which all your result will be compared against.
- The instructors have provided code to covert the sparse matrix to CSR format.
- For Part 1, implement *SpMV_CSR* which calculates the SpMV operation using the CSR formatted matrix.
- For Part 2, convert the CSR data structure to COO format.
- For Part 3, use this COO format to calculate the SpMV operation.
- **DO NOT CHANGE ANY OF THE INSTRUCTOR PROVIDED CODE.** That is, implement the functions using the provided parameters only. Do not add, remove, or modify the data structures or code in any way.
- If everything has been implemented correctly, you should see a message that says you have 0 incorrect answers.

Note that you may choose not to include the extra information printed by the skeleton code in your final solution. These functions were provided just to help you get started in the right (or wrong) direction.

When you have completed your assignment, verify it on ix-dev and then rename the file to `hw03_spmv_answer.c`, then add, commit, and push the file to your Bitbucket repository.

Extra credit will be given to those who use OpenMP to paralellize your code. 2 points will be given for parallelizing *SpMV_CSR*. Parallelizing the other functions will be tricky - think about it and provide answers as to why it can't easily be parallelized. Correct **and detailed** suggestions will be

given another 2 extra points (i.e., single word answers or short, one sentence answers will not be given points).

Have fun with your assignment and don't hesitate to post questions on Piazza if something is ambiguous.