

Lab 9 Matrix Operations

Song Liu (song.liu@bristol.ac.uk)

GA 18, Fry Building,

Microsoft Teams (search "song liu").

Homework

Today's lab is directly related to the CW1.

1. Download the lab file and open `ab1234.c`
2. This file contains solutions to the previous lab questions.
Please have a look and make sure you understand the code.

Homework 1

- In the previous lab, you have written a function `get_elem` that helps you retrieve the `i,j` the element of a matrix `M`.
- Now, similarly, write a `set_elem` function which assigns value `val` to the `i,j`-th element in matrix `M`. The declaration of `set_elem` is already given to you in the skeleton code.
 - ```
set_elem(M, 1, 2, 10);
```
  - This will assign value 10 to the `1,2` th element of `M`.
  - We are using **zero-based indexing**.
- Use the print function to verify the correctness of `set_elem` and `get_elem` function that you have written.

## Homework 2 (Submission)

- In the same file, write another function called `mul`. `mul` function takes three input arguments `A`, `B` and `C` which are all `matrix` structures. It returns no output.
- `mul` computes matrix multiplication between matrix `A` and `B`. The outcome will be stored in matrix `C`.

# Homework 2 (Submission)

## Hint:

- How is the `i,j` th element of `c` computed in a matrix multiplication?
  - Recall the vector dot product exercise in the last week. Do you see any similarities?
  - Once you know how to calculate the `i,j` th element, do it for all `i,j` using a nested `for` loop.
- Call `mul` using provided test cases, print out `A,B` and `c` and check for the correctness of your implementation.

## Homework 2 (Submission)

Write additional code in your `mul` function, so it checks if `A`, `B` and `C` have correct sizes.

- If the matrices do not have correct sizes, print out an error message and exit `mul` function immediately.
- What will happen if matrices do not have correct dimensions, and you did not perform the check?

## Homework 3 (Challenge)

- Write a function called `transpose`. It takes a `matrix` input `A`, and returns a matrix structure.
  - The function returns the transpose of matrix `A`.
- Come up with your own test cases and check the correctness of your implementation.

## Homework 3 (Challenge)

- Now, suppose someone wants to compute  $C = AA^T$ . He writes the program

```
mult(A, transpose(A), C);
```

- Will this benign looking code produce the desired output?
- What is the potential danger of writing code this way?
- How do you fix it?



## Homework 3 (Challenge)

- Hint: add a new variable `transpose` to the matrix struct, indicating whether this matrix has been transposed.
- Hint: Rewrite `get_elem` and `set_elem` functions so that it takes the transposition of the matrix into account.
- Hint: Now, rewrite `transpose` function (it should contain only one line of code).
- You shouldn't need to change any other code and all other function you wrote should still work correctly.

## Homework 3 (Challenge)

- Rename your code and submit to blackboard.

# Fun fact

In modern day operating system, matrix operations are already part of the OS. These matrix operating functions are collectively called **BLAS (Basic Linear Algebra Subprograms)**.

For example, the matrix multiplication function in BLAS has the following declaration:

```
void cblas_sgemm(const enum CBLAS_ORDER Order,
 const enum CBLAS_TRANSPOSE TransA,
 const enum CBLAS_TRANSPOSE TransB,
 const int M, const int N,
 const int K, const float alpha, const float *A,
 const int lda, const float *B, const int ldb,
 const float beta, float *C, const int ldc);
```

These functions declarations are implemented for a varieties of computational devices (including PC, Mac, Android, iOS).

These functions are the workhorses behind modern machine learning software such as Numpy and Tensorflow.