EE231002 Introduction to Programming

Lab07. Matrix Determinants

Due: Nov. 10, 2018

Given an $N \times N$ square matrix $A_{i,j}, 1 \leq i, j \leq N$, then the determinant can be defined by the Leibniz formula as

$$\det(A) = \sum_{\sigma \in S_N} \text{sgn}(\sigma) \prod_{i=1}^{N} A_{i,\sigma_i}. \tag{7.1}$$

where S_N is the set of all permutations of $\{1, 2, \cdots, N\}$, σ is one possible permutation in S_N, and σ_i is the ith element of the permutation σ. In Lab 5, the Pandita algorithm has been introduced. Given a permutation $\sigma^{(m)}$, the Pandita algorithm generates the next lexicographic permutation $\sigma^{(m+1)}$ with $\sigma^{(1)} = \{1, 2, \cdots, N\}$. The function $\text{sgn}(\sigma^{(m)})$ is defined as the following.

$$\text{sgn}(\sigma^{(m)}) = \begin{cases} 1, & \text{if } m = 1, \\ (-1)^t \times \text{sgn}(\sigma^{(m-1)}), & \text{otherwise.} \end{cases} \tag{7.2}$$

where t is the number of swaps needed for the Pandita algorithm to generate the next permutation. An example of $N = 3$ case is listed in the following table.

S_N	σ	t	$\text{sgn}(\sigma^{(m)})$	product
$\sigma^{(1)}$	1 2 3	0	1	$A_{1,1}\ A_{2,2}\ A_{3,3}$
$\sigma^{(2)}$	1 3 2	1	-1	$A_{1,1}\ A_{2,3}\ A_{3,2}$
$\sigma^{(3)}$	2 1 3	2	-1	$A_{1,2}\ A_{2,1}\ A_{3,3}$
$\sigma^{(4)}$	2 3 1	1	1	$A_{1,2}\ A_{2,3}\ A_{3,1}$
$\sigma^{(5)}$	3 1 2	2	1	$A_{1,3}\ A_{2,1}\ A_{3,2}$
$\sigma^{(6)}$	3 2 1	1	-1	$A_{1,3}\ A_{2,2}\ A_{3,1}$

In this assignment, you need to write a C program to calculate the determinant of an $N \times N$ square matrix using Equations (7.1) and (7.2). Since the Pandita algorithm has been introduced in Lab05, you are requested to write a function to generate the next lexicographic permutation using that algorithm. The declaration of Pandita algorithm is as following.

```
int Pandita(int A[N]);
```

The function takes the permutation array `A[N]` as input and rearranges it for the next permutation. It then returns $\text{sgn}(\sigma^{(m)})$ as the output. If it reach the end of the permutation, then it returns 0 to terminate the determinant calculation. To facilitate writing of this assignment, the size of the matrix N should be defined as a macro.

```
#if !defined(N)
#define N 3
#endif
```

Twelve matrices with various dimensions have been provided for you to test your program. They are `mat1.in`, `mat2.in`, $\cdots$, `mat12.in`. You should open each file to find the dimension of the matrix and then compile your program with the right dimension as

```
$ gcc -DN=3 lab07.c
$ ./a.out < mat1.in
```

The last line uses the `unix` input redirection method to read input directly from the file `mat1.in`. In this way, we do not need to retype the matrix every time we execute the program. Example program compilation and execution is shown below.

```
$ gcc -DN=3 lab07.c
$ ./a.out < mat1.in
Input matrix is
  1 2 3
  4 5 6
  7 8 9
Det = 0
```

Notes.

1. Create a directory **`lab07`** and use it as the working directory.
2. Name your program source file as **`lab07.c`**.
3. The first few lines of your program should be comments as the following.

   ```
   /* EE231002 Lab07. Matrix Determinants
      ID, Name
      Date:
   */
   ```

4. After you finish verifying your program, you can submit your source code by

   ```
   $ ~ee2310/bin/submit lab07 lab07.c
   ```

   If you see a "submitted successfully" message, then you are done. In case you want to check which file and at what time you submitted your labs, you can type in the following command:

   ```
   $ ~ee2310/bin/subrec lab07
   ```

   It will show the last few submission records.