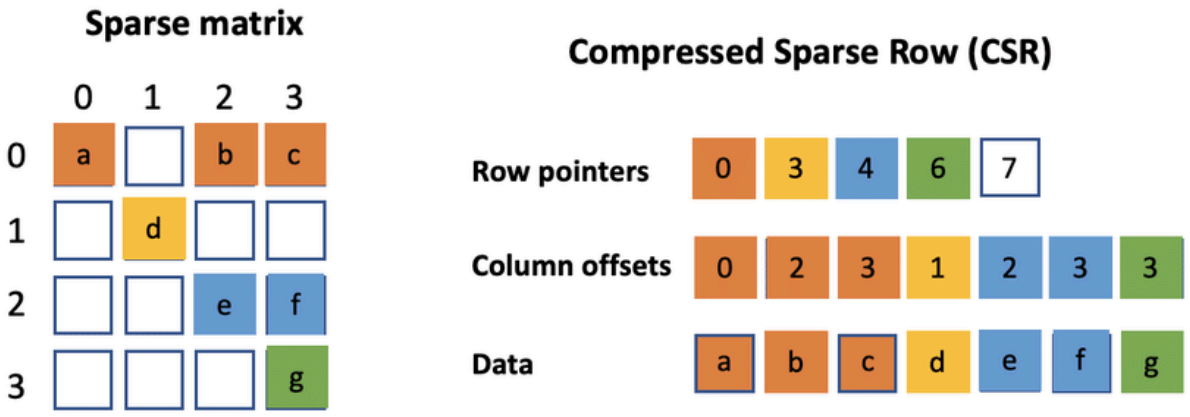
Assignment 3

When the matrices are sparse, it makes sense to efficiently store them in alternate representations than 2-D arrays. One such representation is Compressed Sparse Row (CSR) as shown below:



It stores the matrix as three 1-D arrays:

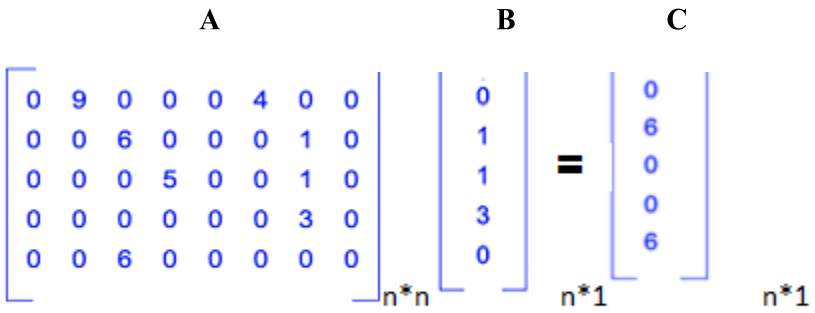
- ***Data:*** stores all non-zero elements in the row-wise order
- ***Column offsets:*** stores the column number of these non-zero elements
- ***Row pointers array:*** i^{th} element of this array stores the cumulative number of non-zero elements upto (not including) the i-th row. The first element is always zero, and hence it is of size n+1, where n is the number of rows in the matrix.

By putting subsequent nonzeros of the matrix rows in contiguous memory locations, the execution time is also reduced significantly for some computations.

(P.T.O)

Problem Statement

Write a program using pthreads library for Sparse Matrix Multiplication (SpMM), which multiplies a sparse matrix (in CSR format) with a dense Vector to generate a vector as output.



Input: i) File 1: sparse matrix in COO format

ii) File 2: A dense vector

iii) Number of threads (as input from user)

Output: Vector C

Instructions:

1. Read **inputfile.mtx**
   a. Convert it into **CSR format** and store it as input **A**
   b. Print #Rows, #Cols, #Non-Zeroes, #threads, and then the matrix.
2. Read **vector.txt** file as another input **B.**
3. Compute **A*B** sequentially and store output in C1 and print C1.
4. Compute **A*B** in parallel using pthreads library and store output in C2, and print C2. **[Algorithm to be used:** If #threads =t, each thread computes n/t elements of C2 (Naïve algorithm) **]**
5. Compare the results of C1 and C2 to establish the correctness.
6. Vary the number of threads (1, 2, 4, 8, 16, 32) and compare the timings with the sequential code.
7. Report any other observations.

(P.T.O)

Note

1. DO NOT PLAGIARISE! Any form of plagiarism will result in immediate deduction of marks for the assignment and may also invite additional penalties in Mid-semester, End-semester, and Internal evaluations.
2. **Every program has to be accompanied with a short and succinct report (in the beginning of the source code as comments**), stating the following details:
    a. - Name, Roll Number
    b. - Description of the experimental set-up (configuration of the computer you used, Compiler used, compilation flags used) – 1-2 Lines.
    c. - a brief summary (1-2 lines) of your results
    d. - Any other remarks/observations (if required)
3. **You need to submit the properly commented and indented C/C++ file (with your roll no. as the file name)** on google classroom on or before **Monday, 18th August, 2025**. The report (commented) also has to be included in the same file.

****