TRUBA
Turkish Science e-Infrastructure

EURO

COMPUTHON 2022/1

TÜBİTAK ULAKBİM

ORTA DOĞU TEKNİK ÜNİVERSİTESİ
MIDDLE EAST TECHNICAL UNIVERSITY

Sabancı Üniversitesi

Computhon Schedule

May 4th (Wednesday, Noon):

Registration deadline

May 5th (Thursday):

Hackathon launch (online)

May 10th (Tuesday, Noon):

Last day of solution submission (online through Computhon webpage)

May 11th-12th (Wednesday-Thursday):

Solution presentations & final evaluation (face-to-face and online)

May 13th (Friday):

Announcement of winners & award ceremony (face-to-face and online)

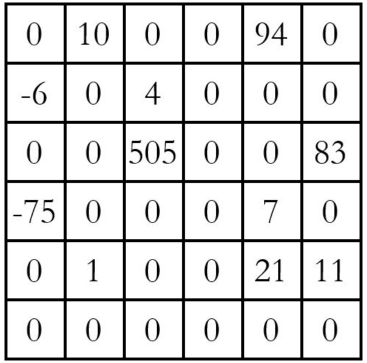
Today’s Program

- Tensors, fibers, and problem definition
- Submission specifications
- Baseline

Tensors

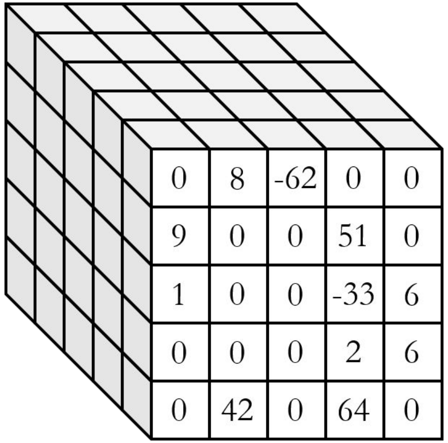
Tensors are multidimensional arrays, i.e., generalizations of matrices.

- The **order** of a tensor is the number of dimensions (or modes).



0	10	0	0	94	0
-6	0	4	0	0	0
0	0	505	0	0	83
-75	0	0	0	7	0
0	1	0	0	21	11
0	0	0	0	0	0

order-2 tensor



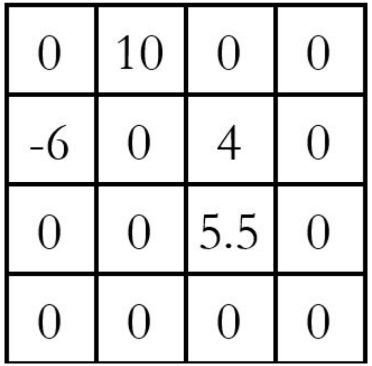
0	8	-62	0	0
9	0	0	51	0
1	0	0	-33	6
0	0	0	2	6
0	42	0	64	0

order-3 tensor

Tensors

Tensors are often stored in-memory in the coordinate (COO) format.

- Coordinate matrix of size (order x #non-zeros)
- The coordinates of each non-zero is a column.
- Each row represents a dimension (mode).
- Values often stored separately (for data type reasons).



0	10	0	0
-6	0	4	0
0	0	5.5	0
0	0	0	0

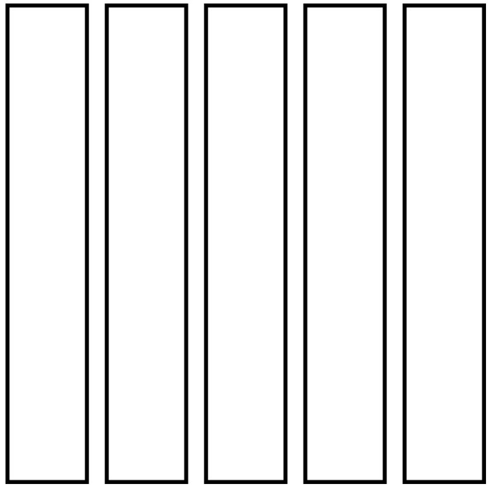
```
COO =   [0,    1,    1,    2]
        [1,    0,    2,    2]

vals = [10, -6,    4,     5.5]
```

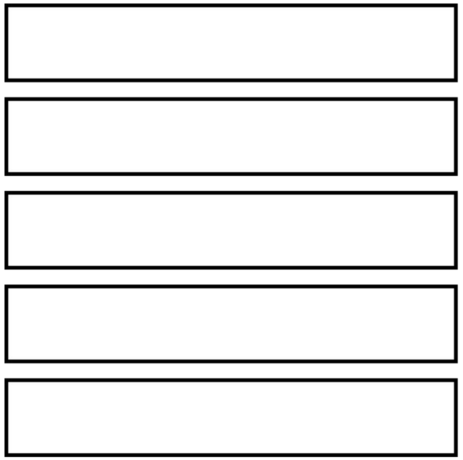
Fibers

Fibers are vectors within tensors defined by fixing every dimension’s index except one (e.g., rows and columns in matrices).

T[:][j]
(columns)

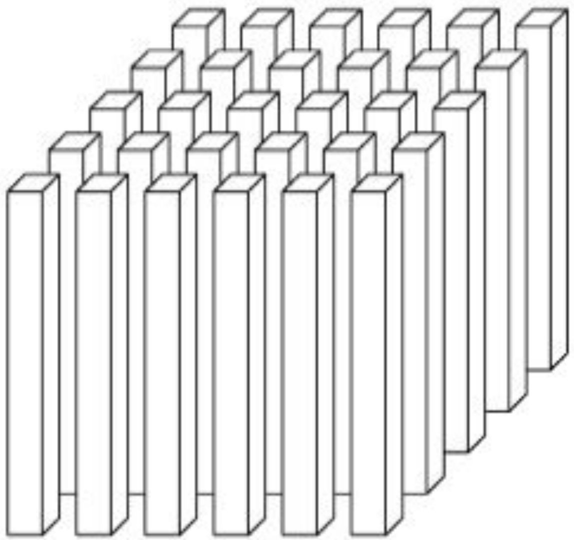


T[i][:]
(rows)

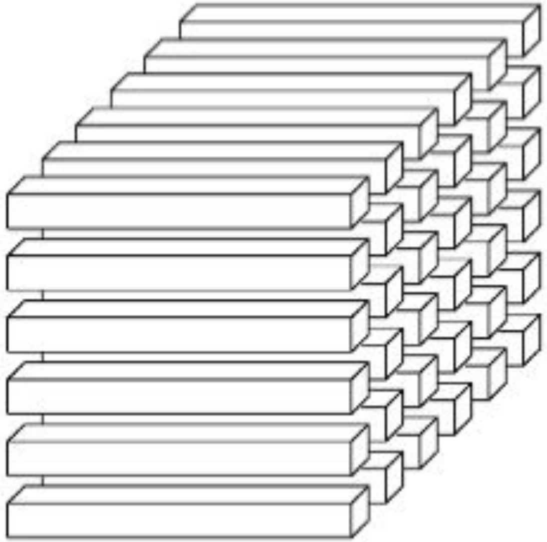


Fibers

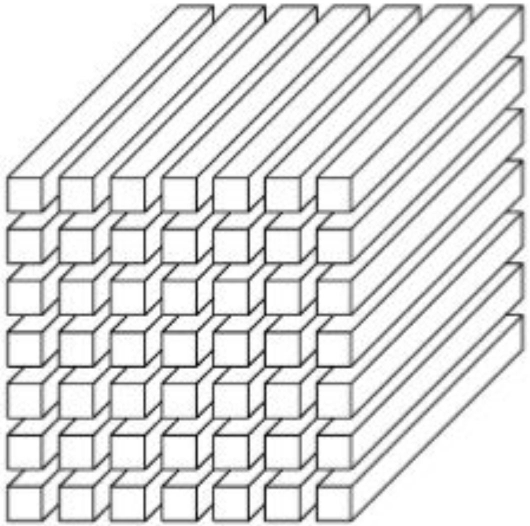
T[:][j][z]
(columns)



T[i][:][z]
(rows)

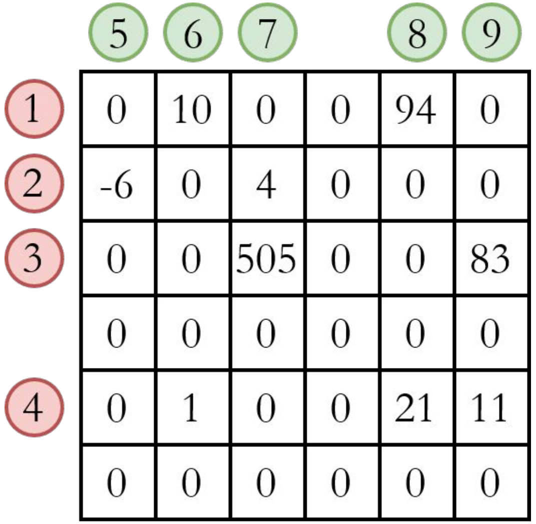


T[i][j][:]
(tubes)



Problem definition

- Given a tensor, count the number of fibers that contain at least one non-zero value.



$$\begin{array}{c|cccccc}
 & 5 & 6 & 7 & & 8 & 9 \\
\hline
1 & 0 & 10 & 0 & 0 & 94 & 0 \\
2 & -6 & 0 & 4 & 0 & 0 & 0 \\
3 & 0 & 0 & 505 & 0 & 0 & 83 \\
 & 0 & 0 & 0 & 0 & 0 & 0 \\
4 & 0 & 1 & 0 & 0 & 21 & 11 \\
 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

Specifications

- You are not responsible for reading the tensor files (.tns).
- You will process a tensor stored in the COO format.
- The values array will not be given with the COO matrix (and is not needed).

Expected output: a single integer which is the number of non-zero fibers.

Provided data

- Main program + baseline fiber counting function.
- Example tensor inputs.
- Sample SLURM scripts for execution.
- OpenMP, MPI, and CUDA training material.

Link: https://github.com/SU-HPC/Computhon2022-1

Categories

- Single-node: Parallel computation on multi-core CPUs.
- Single-node: Parallel computation on multi-core CPUs and multiple GPUs.
- Multi-node: Distributed computing on multiple nodes + CPUs and GPUs

Grading criteria

- Test tensors have at most 5 billion non-zeros and 10 modes.
- The dimension on any mode < UINT_MAX i.e. biggest number that can be stored in an unsigned integer.
- Average speedup on all of our testing tensors (will not be shared with you!)
- Best average speedup wins.
- In case your code times-out/returns incorrect results, you will be penalized (will count as a 0.5x speedup).

Submission format

- For each category:
  - One code and make file.
  - The code file must have **an identical main function** to the one in the given solution file.
    - If you think that you cannot build the solution without modifying the main function, you need to get permission from us first.
  - A README.txt file for each category to give compilation and running details.

Baseline

```
/*
Parameters:
* [In]  nnz: unsigned int - number of non-zeros
* [In]  order: unsigned int - number of dimensions
* [In]  dimensions: unsigned int* - an array of size `order` containing the dimension on each mode
* [In]  coord: unsigned int** - 2D structure of dimensions `order` x `nnz` with each column
representing a non-zero
        ex: coord[i][j] = the i'th dimension value of the j'th non-zero in the tensor

Returns:
* unsigned long long - number of non-zero fibers in the tensor

*/
unsigned long long naive_count_coo(unsigned int nnz, unsigned int order, unsigned int* dimensions,
unsigned int ** coord);
unsigned long long naive_count_coo(unsigned int nnz, unsigned int order, unsigned int* dimensions, unsigned int **
coord){
    unsigned long long total_fibers = 0;
    // Go over each mode (find fibers on that mode)
    for (int m = 0; m < order; m++){
        // modes other than `m` (dimensions that will be constant for fibers on mode `m`)
        vector<int> modes_to_check;
        for (int x = 0; x < order; x++) {
            if (x == m) continue;
            modes_to_check.push_back(x);
        }
        // Set containing all the fibers on mode `m` based on the constant indices of each fiber
        unordered_set<vector<unsigned int>, vector_hash<unsigned int>, vector_equal<unsigned int>> fibers;
        vector<unsigned int> one_nnz(order-1);
        // Determine the fiber that each non-zero belongs to and add it to the set
        for (unsigned int i = 0; i < nnz; i++){
            // Construct the fiber ID using its dimensions on all modes except `m`
            for (int o = 0; o < order-1; o++){
                one_nnz[o] = coord[modes_to_check[o]][i];
            }
            fibers.insert(one_nnz);
        }
        // Add the fiber count in mode `m` to the total fiber count
        total_fibers+=fibers.size();
    }
    return total_fibers;
}
```