GENERIC PROGRAMMING PROJECT

TITLE: SPARSE MATRIX CONTAINER

ZENKAR S PES1201701532
RAVENDRA PES1201700706

Sparse matrix is a matrix with more zero values than non-zero values. Sparse matrix operations are common in machine learning and computer graphics.
When storing and manipulating sparse matrices on a computer, it is beneficial and often necessary to use specialized algorithms and data structures that take advantage of the sparse structure of the matrix.

Space complexity of 2D array = theta(M*N)
Space complexity of sparse matrix = theta(nnz)

Consider a MxN matrix with nnz number of non-zero values

OPERATION	TIME COMPLEXITY **2D matrix**	TIME COMPLEXITY **Sparse matrix container**
Access	O(1)	O(log(N)) : map O(1) : unordered map
Find	O(M*N)	O(nnz)
Transpose	O(M*N)	O(N+nnz)
Matrix multiplication A(MxN) * B(NxP)	O(MNP)	O(P*nnz(A) + M*nnz(B))
Matrix Addition	O(M*N)	O(nnz)

Iterator: Forward iterator

Implementation:
template<class T ,class Container = std::map<T>> T is the value type
Container can be std::map or std::unordered_map

Use Case:
1. Perform matrix operations like multiplication and addition on sparse matrices with better time complexity than a 2D array