**[Sparse Matrix Multiplication](https://leetcode.com/problems/sparse-matrix-multiplication/)**

**public** **class** SparseMultiplication {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int**[][] A = {{1, 0, 0} , {-1, 0, 3}};

**int**[][] B = {{7, 0, 0}, {0, 0, 0 } , {0, 0, 1}};

**int**[][] multiply = *multiply*(A , B);

**for**(**int** i = 0 ; i < multiply.length ; i++) {

**for**(**int** j = 0 ; j < multiply[0].length ; j++) {

System.***out***.print(multiply[i][j]+ " ");

}

System.***out***.println();

}

}

**public** **static** **int**[][] multiply(**int**[][] A, **int**[][] B) {

**if**(A == **null** || B == **null** || A.length == 0 || B.length == 0) {

**return** **new** **int**[][] {};

}

**int** m = A.length;

**int** n = B[0].length;

**int** mn = B.length;

**int**[][] multiply = **new** **int**[m][mn];

**for**(**int** i = 0 ; i < m ; i++) {

**for**(**int** j = 0 ; j < n ; j++) {

**for**(**int** k = 0 ; k < mn ; k++) {

multiply[i][j] += A[i][k] \* B[k][j];

}

}

}

**return** multiply;

}

}

Time complexity : O(m \* n) ,n is Number elements in array A and m is number of elements in B

Space Complexity : O(m \* n) , n is Number elements in array A and m is number of elements in B