**ASSIGNMENT 2**

**AIM:**

Accept conventional matrix and convert it into sparse matrix. Implement simple and fast transpose algorithms on sparse matrix.

**OBJECTIVE:**

The objective of this program is to learn how the sparse matrix works, and to learn two types of transpose: Simple & Fast and implementation of it on the Sparse matrix.

**THEORY:**

**Sparse Matrix:**

Sparse Matrix is that matrix which has which has a very few non zero elements.

Representation of sparse matrix saves alot of memory space.

**Transpose of a matrix:**

A matrix obtained from a given matrix by interchanging each row and the Corresponding column. It has two types 1)Simple & 2) Fast.

**CODE:**

#include<iostream>

using namespace std;

int main()

{

int row ,col,i,j,nz=0;

cout<<"Enter number of rows and columns: ";

cin>>row>>col;

int mat1[row][col];

cout<<"\nEnter the values: ";

for(i=0;i<row;i++)

{

for(j=0;j<col;j++)

{

cin>>mat1[i][j];

if(mat1[i][j]!=0)

nz++;

}

}

int sparsemat[nz+1][3],index=1;

sparsemat[0][0]=row;

sparsemat[0][1]=col;

sparsemat[0][2]=nz;

for(i=0;i<row;i++)

{

for(j=0;j<col;j++)

{

if(mat1[i][j]!=0)

{

sparsemat[index][0]=i;

sparsemat[index][1]=j;

sparsemat[index][2]=mat1[i][j];

index++;

}

}

}

cout<<"The resultant sparse matrix is :\n";

for(i=0;i<index;i++)

{

for(j=0;j<3;j++)

{

cout<<" "<<sparsemat[i][j];

}

cout<<"\n";

}

int transpose1[nz+1][3],index2=1;

transpose1[0][0]=col;

transpose1[0][1]=row;

transpose1[0][2]=nz;

for(int pass=0;pass<col;pass++)

{

for(index=1;index<=nz;index++)

{

if(sparsemat[index][1]==pass)

{

transpose1[index2][0]=sparsemat[index][1];

transpose1[index2][1]=sparsemat[index][0];

transpose1[index2][2]=sparsemat[index][2];

index2++;

}

}

}

cout<<"The resultant transpose matrix using simple transpose is :\n";

for(i=0;i<=nz;i++)

{

for(j=0;j<3;j++)

{

cout<<" "<<transpose1[i][j];

}

cout<<"\n";

}

int transpose[nz+1][3];

int countt[col],stindex[col];

transpose[0][0]=col;

transpose[0][1]=row;

transpose[0][2]=nz;

for(i=0;i<col;i++)

countt[i]=0;

for(index=1;index<=nz;index++)

countt[sparsemat[index][1]]++;

stindex[0]=1;

for(i=1;i<col;i++)

stindex[i]=stindex[i-1]+countt[i-1];

for(index=1;index<=nz;index++)

{

index2=stindex[sparsemat[index][1]];

stindex[sparsemat[index][1]]++;

transpose[index2][0]=sparsemat[index][1];

transpose[index2][1]=sparsemat[index][0];

transpose[index2][2]=sparsemat[index][2];

index2++;

}

cout<<"The resultant transpose matrix using fast transpose is :\n";

for(i=0;i<=nz;i++)

{

for(j=0;j<3;j++)

{

cout<<" "<<transpose[i][j];

}

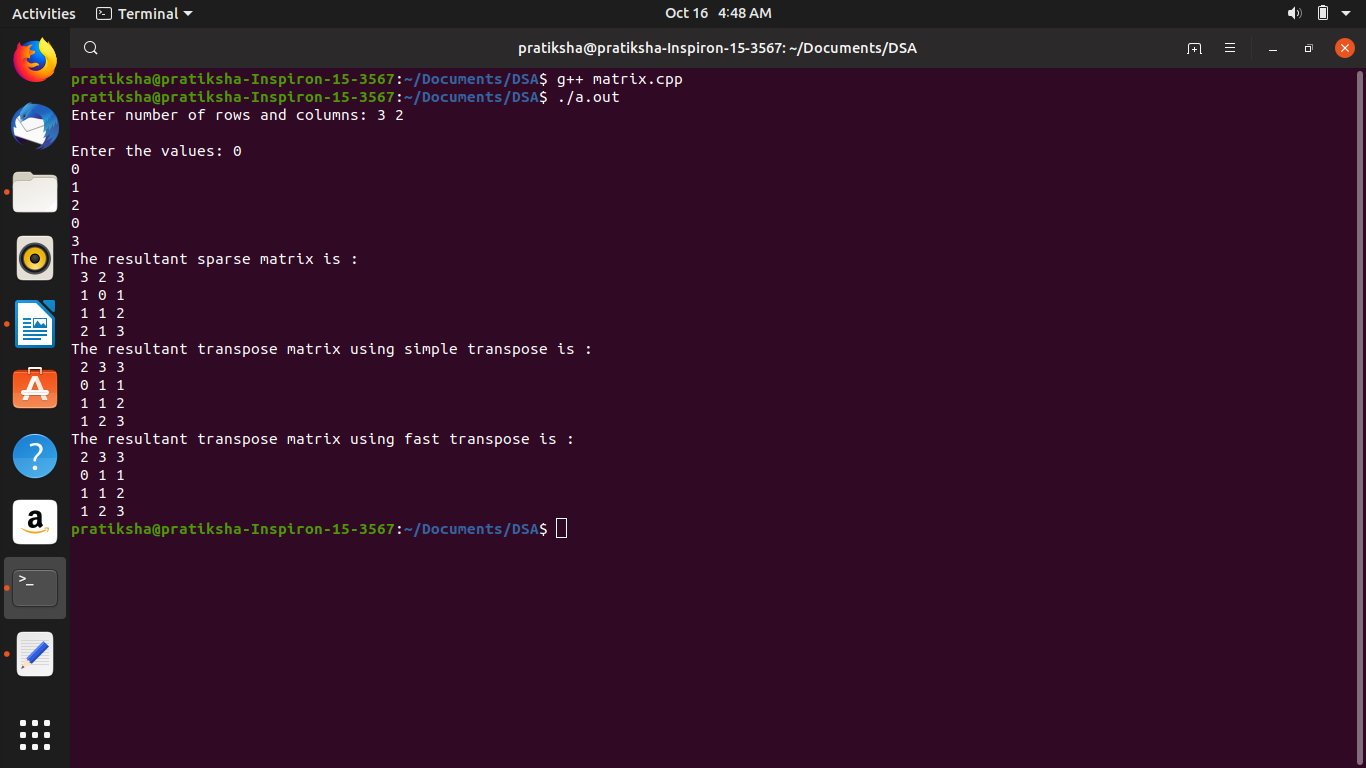
cout<<"\n";

}

return 0;

}

**OUTPUT:**



**COMPLEXITY:**

**Simple Transpose**: O(n^2)

**Fast Transpose**: O(n)

**CONCLUSION:**

In this program we learnt the importance of a sparse matrix and how to find its transpose. We also learnt that there are two types of sparse matrix transpose 1) Simple and 2) Fast.

Last modified: 7:06 am