

Desarrollar los algoritmos en C# para las siguientes operaciones

$C = A_{n \times m} + B_{n \times m}$ usando matrices normales

```
public void MatrixSum(int[,] matA, int[,] matB, int[,] matC)
```

```
{  
    if ((matA.GetLength(0) == matB.GetLength(0)) &&  
        (matA.GetLength(1) == matB.GetLength(1)))  
    {  
        int rows = matA.GetLength(0);  
        int columns = matA.GetLength(1);  
  
        for (int i = 0; i < rows; i++)  
        {  
            for (int j = 0; j < columns; j++)  
            {  
                if (matA[i, j] > 0 || matB[i, j] > 0)  
                    matC[i, j] = matA[i, j] + matB[i, j];  
                else  
                    matC[i, j] = 0;  
            }  
        }  
    }  
}
```

$C = A_{n \times m} + B_{n \times m}$ usando matrices sparse

```
public clsSpar SumaMatrizSpar(clsSpar A, clsSpar B)

{

    clsSpar C = new clsSpar();

    C.V[0, 0] = A.n;

    C.V[0, 1] = A.m;

    C.Cant = 1;

    if (A.Cant > B.Cant)

    {

        for (int i = 1; i <= A.Cant; i++)

        {

            for (int j = 1; j <= B.Cant; j++)

            {

                if ((A.V[i, 0] == B.V[j, 0]) && (A.V[i, 1] == B.V[j, 1]))

                {

                    if ((A.V[i, 2] + B.V[j, 2]) != 0)

                    {

                        C.V[C.Cant, 0] = A.V[i, 0];

                        C.V[C.Cant, 1] = A.V[i, 1];

                        C.V[C.Cant, 2] = A.V[i, 2] + B.V[j, 2];

                        C.Cant++;

                    }

                    break;

                }

                if (j == B.Cant)

                {


```

```

        C.V[C.Cant, 0] = A.V[i, 0];

        C.V[C.Cant, 1] = A.V[i, 1];

        C.V[C.Cant, 2] = A.V[i, 2];

        C.Cant++;

    }

}

}

}

else

{

    for (int i = 1; i <= B.Cant; i++)

    {

        for (int j = 1; j <= A.Cant; j++)

        {

            if ((B.V[i, 0] == A.V[j, 0]) && (B.V[i, 1] == A.V[j, 1]))

            {

                if ((B.V[i, 2] + A.V[j, 2]) != 0)

                {

                    C.V[C.Cant, 0] = B.V[i, 0];

                    C.V[C.Cant, 1] = B.V[i, 1];

                    C.V[C.Cant, 2] = B.V[i, 2] + A.V[j, 2];

                    C.Cant++;

                }

            }

        }

    }

}

```

```
        break;

    }

    if (j == A.Cant)

    {

        C.V[C.Cant, 0] = B.V[i, 0];

        C.V[C.Cant, 1] = B.V[i, 1];

        C.V[C.Cant, 2] = B.V[i, 2];

        C.Cant++;

    }

}

}

}

C.V[0, 2] = C.Cant - 1;

return C;

}
```