**Bipartite Graph:**

#include<bits/stdc++.h>

using namespace std;

int M,N;

bool Graph[10][10];

bool bpm(bool Graph[10][10], int u,

bool seen[], int matchR[])

{

for (int v = 0; v < N; v++)

{

if (Graph[u][v] && !seen[v])

{

seen[v] = true;

if (matchR[v] < 0 || bpm(Graph, matchR[v],seen, matchR))

{

matchR[v] = u;

return true;

}

}

}

return false;

}

int maxBPM(bool Graph[10][10])

{

int matchR[N];

memset(matchR, -1, sizeof(matchR));

int result = 0;

for (int u = 0; u < M; u++)

{

bool seen[N];

[3:03 pm, 29/06/2022] +91 95918 87988: memset(seen, 0, sizeof(seen));

if (bpm(Graph, u, seen, matchR))

{

result++;

}

}

return result;

}

int main()

{

cout << "Enter the number of elements in the first and the second set" <<

endl;

cin >> M >> N;

cout << "Enter the matrix" << endl;

for(int i = 0;i < M;i++)

{

for(int j = 0;j < N;j++)

{

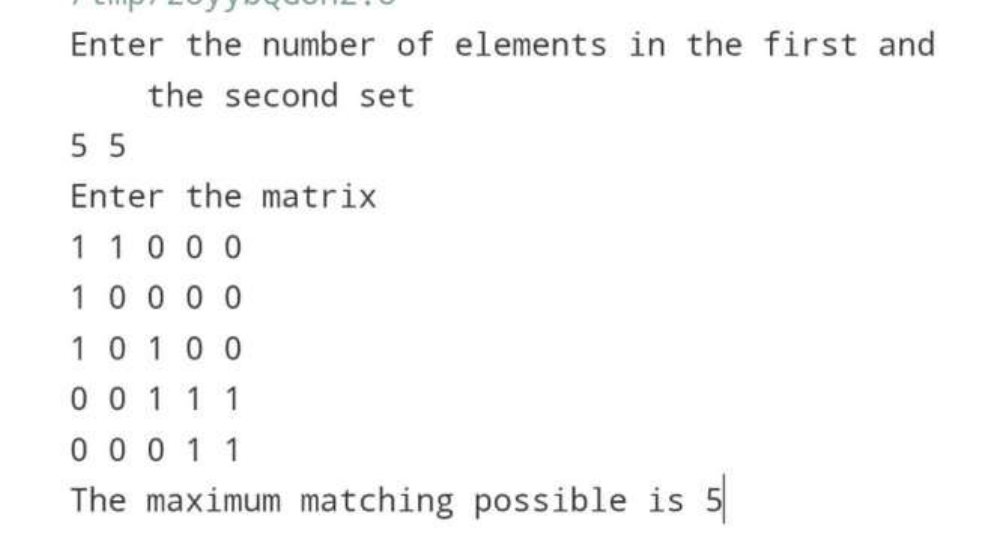
cin >> Graph[i][j];

}

cout << "The maximum matching possible is " << maxBPM(Graph);

return 0;

}



**Matrix Multiplication Chain**:

#include<iostream>

using namespace std;

void Matrix(int d[],int n)

{

int a[n][n],b[n],min,count=0;

bool var[n];

string eq="(";

for(int i=0;i<n;i++)

{

var[i]=false;

for(int j=0;j<n-i;j++)

{

a[j][j+i]=9999;

for(int k=j;k<j+i;k++)

{

min=a[j][k]+a[k+1][j+i]+(d[j]\*d[k+1]\*d[j+i+1]);

if(min<a[j][j+i])

{

a[j][j+i]=min;

if(j==0)

b[j+i]=k;

}

}

if(i==0)

a[j][j+i]=0;

}

}

for(int i=n-1;i>0;i--)

{

var[b[i]+1]=true;

}

for(int i=1;i<n;i++)

{

if(var[i])

{

eq+="a"+to\_string(i)+")";

count++;

}

else

eq+="a"+to\_string(i);

}

for(int i=0;i<count;i++)

{

eq="("+eq;

}

eq+="a"+to\_string(n)+")";

cout<<endl<<"Minimum Multiplications: "<<a[0][n-1]<<endl;

cout<<"The equation with minimum multiplications is: "<<eq<<endl;

}

int main()

{

int n;

cout<<"Enter the number of matrices\n";

cin>>n;

int a[n][2];

int d[n+1];

cout<<"Enter the size of the matrices in order of row and columns\n";

for(int i=0;i<n;i++)

{

cout<<"Matrix "<<i+1<<endl;

cout<<"Row: ";

cin>>a[i][0];

cout<<"Column: ";

cin>>a[i][1];

}

for(int i=0;i<=n;i++)

{

if(i!=n)

d[i]=a[i][0];

else

d[i]=a[i-1][1];

}

Matrix(d,n);

return 0;

}

