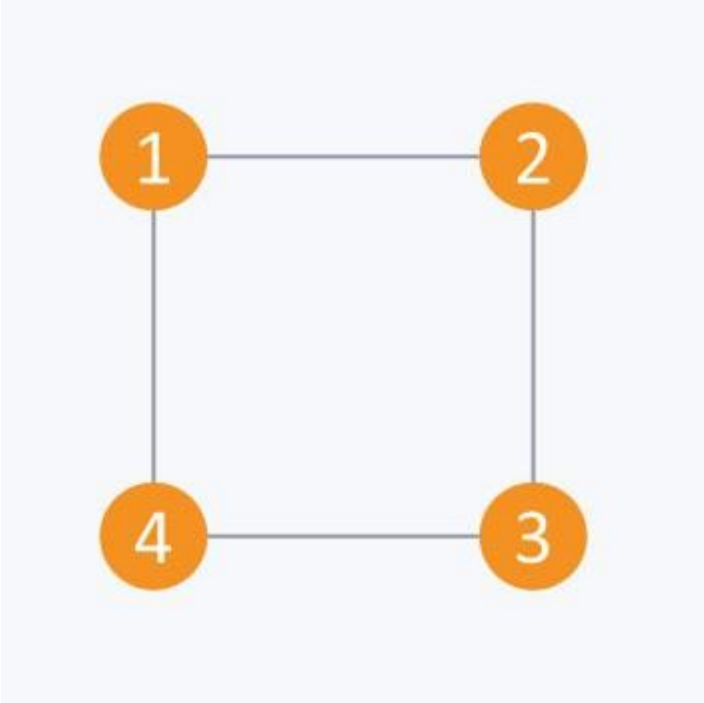
Graph representation using adjacency matrix

Presented by: Adarsh Singh

What is adjacency matrix?

- An adjacency matrix is a **V x V** binary matrix **A** (a binary matrix is a matrix in which the cells can have only one of two possible values - either a 0 or 1). Element $\mathbf{A_{i,j}}$ is 1 if there is an edge from vertex i to vertex j else $\mathbf{A_{i,j}}$ is 0. The adjacency matrix can also be modified for the weighted graph in which instead of storing 0 or 1 in Ai,j we will store the weight or cost of the edge from vertex i to vertex j. In an undirected graph, if $\mathbf{A_{i,j}} = 1$ then $\mathbf{A_{j,i}} = 1$. In a directed graph, if $\mathbf{A_{i,j}} = 1$ then $\mathbf{A_{j,i}}$ may or may not be 1. Adjacency matrix provides **constant time access (O(1))** to tell if there is an edge between two nodes. Space complexity of adjacency matrix is $\mathbf{O(V^2)}$.

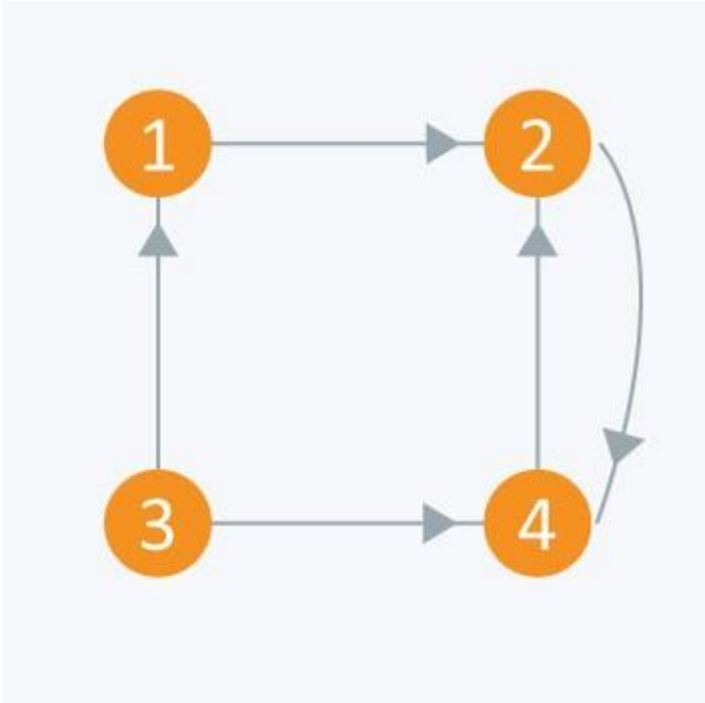
Adjacency matrix(undirected graph)



The adjacency matrix of the graph is:

i/j : **1 2 3 4**
1 : 0 1 0 1
2 : 1 0 1 0
3 : 0 1 0 1
4 : 1 0 1 0

Adjacency matrix(directed graph)



The adjacency matrix of the graph is:

i/j:	**1**	**2**	**3**	**4**
1 :	0	1	0	0
2 :	0	0	0	1
3 :	1	0	0	1
4 :	0	1	0	0

File Edit Selection Find View Goto Tools Project Preferences Help

adja.cpp

```cpp
#include <iostream>

using namespace std;

bool A[10][10];

void initialize()
{
    for(int i = 0;i < 10;++i)
        for(int j = 0;j < 10;++j)
            A[i][j] = false;
}

int main()
{
    int x, y, nodes, edges;
    initialize();       // Since there is no edge initially
    cin >> nodes;       // Number of nodes
    cin >> edges;       // Number of edges
    for(int i = 0;i < edges;++i)
    {
        cin >> x >> y;
        A[x][y] = true;     // mark the edges from vertex x to vertex y
    }
    if(A[3][4] == true)
        cout << "There is an edge between 3 and 4" << endl;
  else
        cout << "There is no edge between 3 and 4" << endl;

  if(A[2][3] == true)
        cout << "There is an edge between 2 and 3" << endl;
  else
        cout << "There is no edge between 2 and 3" << endl;

  return 0;
}
```

Line 19, Column 43 Spaces: 2 C++

Input:

4
5
1 2
2 4
3 1
3 4
4 2

Output:

There is an edge between 3 and 4
There is no edge between 2 and 3

Thanks