SUPERIOR UNIVERSITY

Project

Discrete Mathematics – Final Project

Question

Given the list of edges of simple graph, determine whether the graph is Bipertite?

Group Members

M. Rehan Asghar BSSE 15126

Anam Hassan BSSE 15127

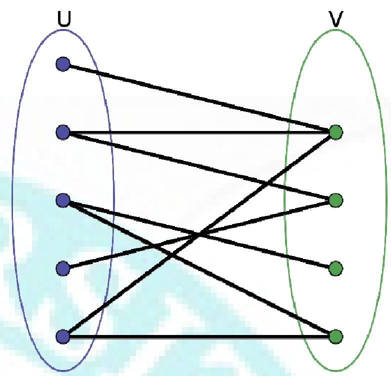
Faizan Ali Khan BSSE 15125

PROFESSOR

Sir Azam (HOD)

Documentation of Bipartite Graph

Definitions:



- **Graph:**

  "Graph is a combination of vertices and edges where set of vertices is non-empty".

- **Types of Graphs:**

  There are following types of graphs.

- Directed Graph.
- Undirected Graph.
- Bipartite Graph.
- Isomorphic Graph.

- **Directed Graph:**

  "The type of graph in which the directions are present"

- **Undirected Graph:**

  "The type of graph in which the directions are not present"

- **Bipartite Graph:**

  "The type of graph in which all the points have same points"

- **Isomorphic Graph:**

  "An isomorphism of graphs G and H is a bijection between the vertex sets of G and H such that any two vertices u and v of G are adjacent in G"

- **Bipartite Graph:**

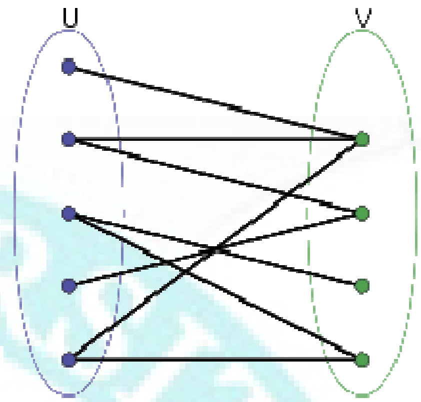
  "It is a type of graph in which all the points have one common point".

- **Detail:**

  In the mathematical field of graph theory, a bipartite graph (or bigraph) is a graph whose vertices can be divided into two disjoint sets U and V (that is, U and V are each independent sets) such that every edge connects a vertex in U to one in V. Vertex set U and V are often denoted as partite sets.

Equivalently, a bipartite graph is a graph that does not contain any odd-length cycles.

Example:



Another example where bipartite graphs appear naturally is in the (NP-complete) railway optimization problem, in which the input is a schedule of trains and their stops, and the goal is to find a set of train stations as small as possible such that every train visits at least one of the chosen stations. This problem can be modeled as a dominating set problem in a bipartite graph that has a vertex for each train and each station and an edge for each pair of a station and a train that stops at that station.

Practical Application in Computer Science:

Bipartite graphs are extensively used in modern coding theory, especially to decode codewords received from the channel. Factor graphs and Tanner graphs are examples of this. A Tanner graph is a bipartite graph in which the vertices on one side of the bipartition represent digits of a codeword, and the vertices on the other side represent combinations of digits that are expected to sum to zero in a codeword without errors. A factor graph is a closely related belief network used for probabilistic decoding of LDPC and turbo codes.

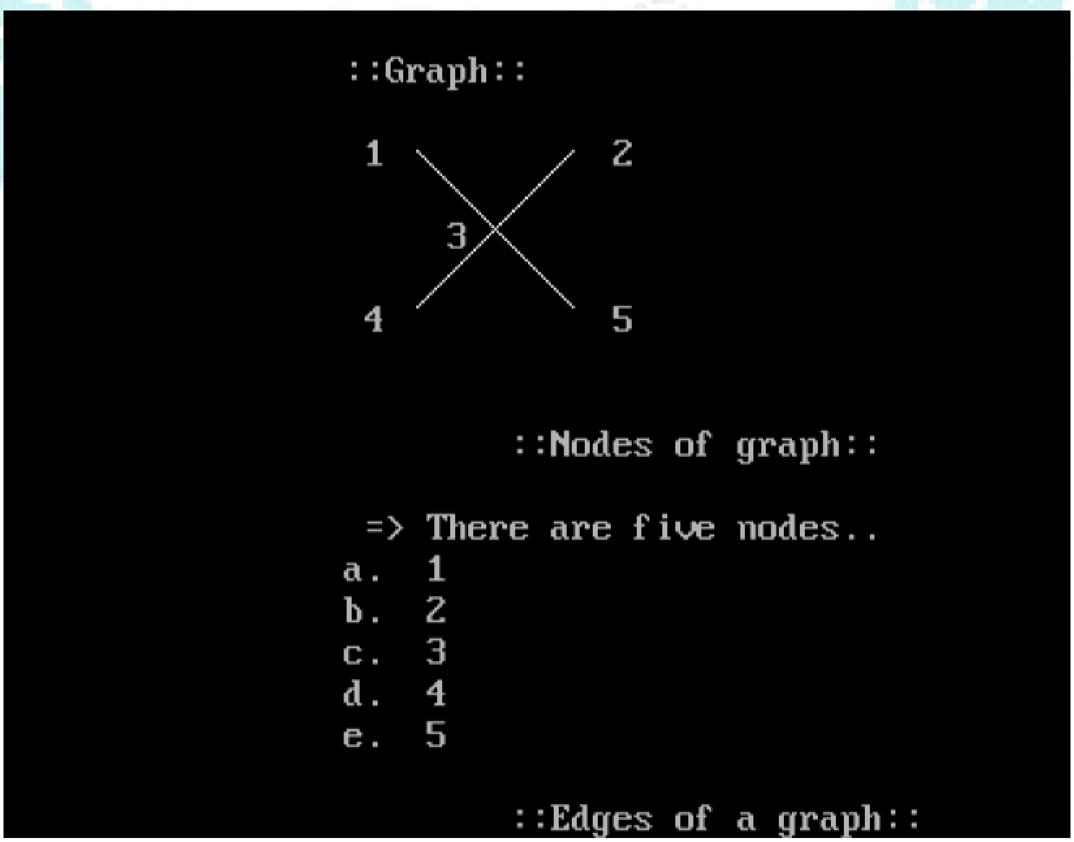
In computer science, a Petri net is a mathematical modeling tool used in analysis and simulations of concurrent systems. A system is modeled as a bipartite directed graph with two sets of nodes: A set of "place" nodes that contain resources, and a set of "event" nodes which generate and/or consume resources. There are additional constraints on the nodes and edges that constrain the behavior of the system. Petri nets utilize the properties of bipartite directed graphs and other properties to allow mathematical proofs of the behavior of systems while also allowing easy implementation of simulations of the system.

Program Coding:

```
#include<graphics.h>
#include<conio.h>
#include<iostream.h>
int main ()
{
  int f,g,h,i;
  int driver=DETECT,mode;
  initgraph(&driver,&mode,"C:\\turboc3\\bgi");
  cout<<"\n\n\t ::Graph::";
  cout<<"\n\n\t  1";
  cout<<"\t      2";
  cout<<"\n\n\t      3";
  cout<<"\n\n\t  4";
  cout<<"\t      5";
  line(100,70,130,100);
  line(130,100,160,70);
  line(130,100,100,130);
  line(130,100,160,130);
  cout<<"\n\n\n\t\t ::Nodes of graph::";
  cout<<"\n\n\t  => There are five nodes..";
  cout<<"\n\t a.  1\n\t b.  2\n\t c.  3\n\t d.  4\n\t e.  5";
  cout<<"\n\n\t\t ::Edges of a graph::";
  cout<<"\n\n\t  => There are four edges..";
  cout<<"\n\n\tEnter the 1st Edge:";
  cin>>f;
  cout<<"\n\tEnter the 2nd Edge:";
  cin>>g;
  cout<<"\n\tEnter the 3rd Edge:";
  cin>>h;
  cout<<"\n\tEnter the 4th Edge:";
```

```
  cin>>i;
  cout<<"\n\t => Edges Of Graph are::";
  cout<<"\n\n\t*  "<<f<<"\n\t*  "<<g<<"\n\t*  "<<h<<"\n\t*  "<<i;
 if(f%10==3&&g%10==3&&h%10==3&&i%10==3)
  {
    cout<<"\n\n\t Graph is Bipartite"<<"\n\n\t Because one point is
common in all edges";
  }
  else
  {
    cout<<"\n\n\t  Graph is not Bipartite"<<"\n\n\t Because there is no
common point";
  }

  getch ();
  return 0;
}
```

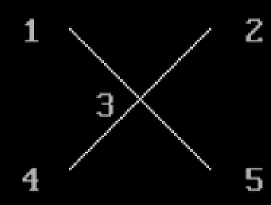
Output:



```
  => There are four edges..

Enter the 1st Edge:13

Enter the 2nd Edge:23

Enter the 3rd Edge:43

Enter the 4th Edge:53

 => Edges Of Graph are::

*  13
*  23
*  43
*  53

 Graph is Bipartite

 Because one point is common in all edges
```

Another Output:

```
::Graph::
```




```
::Nodes of graph::

 => There are five nodes..
a.  1
b.  2
c.  3
d.  4
e.  5

::Edges of a graph::
```

```
 => There are four edges..

Enter the 1st Edge:11

Enter the 2nd Edge:22

Enter the 3rd Edge:44

Enter the 4th Edge:55

 => Edges Of Graph are::

*  11
*  22
*  44
*  55

  Graph is not Bipartite

 Because there is no common point
```