Unit-4

Graph Problem

Page No.
Date

Que- What is graph and their Representation.

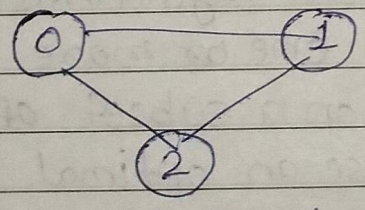
⇒

Graph :-
A graph G can be defined as an ordered set G(V,E) where V(G) represents the set of vertices and E(G) represents the set of edges which are used to connect these vertices.
- vertex - nodes
- Edge - connections between vertices

Types of Graph :-
1) Directed Graphs
2) UnDirected Graphs
3) Weighted Graphs
4) unweighted graphs

Representations of Graph :-

1) Adjacency matrix :-
An adjancency matrix is a square matrix of NXN size where N is the number of nodes in the graph and it is used to represent the connections between edges of graph.



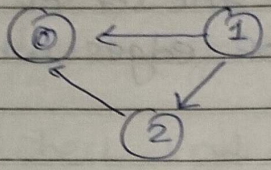
Undirected Graph

*	0	1	2
0	0	1	1
1	1	0	1
2	1	1	0

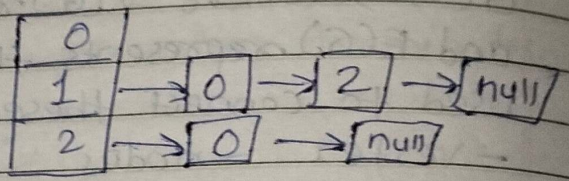
Adjacency matrix

Date

2] Adjacency List :-
An adjacency list is a data structure used to represent a graph where each node in the graph stores a list of its neighbouring vertices.



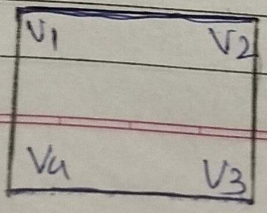
directed graph



Adjacency List.

Imp
Que Explain Working of edge coloring with example.
⇒

Edge coloring of a Graph :-
- In graph theory, edge coloring of a graph is an assignment of "colors" to the edges of the graph so that no two adjacent edges have the same color with an optimal number of colors.
- Two edges are said to be adjacent if they are connected to the same vertex.
- There is no known polynomial time algorithm for edge-coloring every graph with an optimal number of colors.
- Nevertheless, a number of algorithms have been developed that relax one or more of these criteria, they only work on a subset of graphs, or they do not always use an optimal number of colors, or they do not always run in polynomial time.



- Edge colorings are one of several different types of graph coloring problems.
- The above figure of a Graph shows an edge coloring of a graph by the colors blue and gray in which no adjacent edge have the same color.
- Below is an algorithm to solve the edge coloring problem which may not an optimal number of colors :-

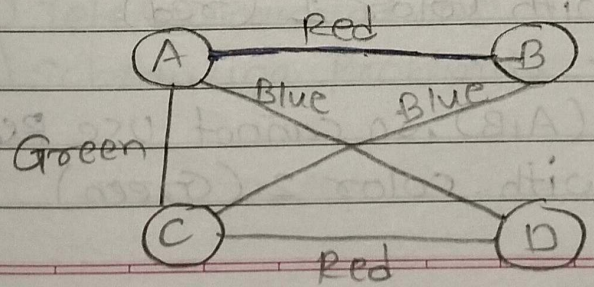
Algorithm :-

1) Use BFS traversal to start traversing the graph.
2) Pick any vertex and give different colors to all of the edges connected to it, and mark those edges as colored.
3) Traverse one of it's edges.
4) Repeat step to with a new vertex until all edges are colored.

- Edge coloring is a type of graph coloring where the goal is to assign colors to the edges of a graph so that no two adjacent edges share the same color.

Example :-

Consider a simple graph G with vertices $V = \{A, B, C, D\}$ and edges $E = \{(A,B), (A,C), (A,D), (B,C), (C,D)\}$



Page No.
Date

Steps of Edge coloring :-

1) Determine the maximum degree (Δ) of the graph :-
- The degree of a vertex is the number of edges connected to it.
- For this graph :-
  - Degree of A is 3 (connected to B, C, D)
  - Degree of B is 2 (connected to A, C)
  - Degree of C is 3 (connected to A, B, D)
  - Degree of D is 2 (connected to A, C)
- The maximum degree Δ is 3.

2) Choose the number of colors :-
- According to Vizing's Theorem, any simple graph can be edge-colored with Δ or $\Delta+1$ colors.
- In our example, we will use $\Delta = 3$ colors.

3) color the edges :-
- start with an arbitrary edge and color it with the First color.
- continue coloring, ensuring no two adjacent edges share the same color.

4) Coloring process :-

a) Edge (A, B) :-
- color it with color 1 (red)

b) Edge (A, C) :-
- Adjacent to (A, B), so cannot use Red
- color it with color 2 (Green)

Page No.
Date

c] Edge (A, D):-
- Adjacent to (A, B) and (A, C) so cannot use Red or Green.
- color it with color 3 (Blue).

d] Edge (B, C):-
- Adjacent to (A, B) and (A, C), already using Red and Green.
- color it with color 3 (Blue)

e] Edge (C, D):-
- Adjacent to (A, C) and (B, C), already using Green and Blue.
- Color it with color 1 (Red).

Final Edge coloring :-
- (A, B) - Red
- (A, C) - Green
- (A, D) - Blue
- (B, C) - Blue
- (C, D) - Red.

- This coloring ensures that no two adjacent edges share the same color.

Applications of edge coloring :-
1) Transport Networks
2) Computer Networks
3) Network Design
4) Games and Puzzles
5) Telecommunications
6) Scheduling Problems.

Page No.
Date

Que- Explain Working of vertex coloring with example.

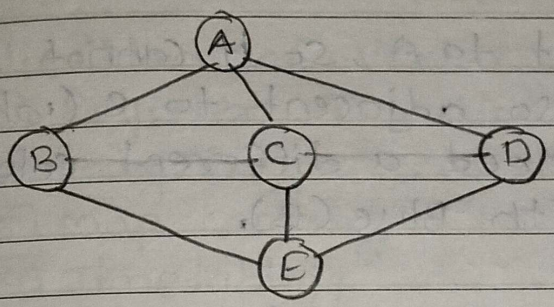
⟹

Vertex coloring :-

- Vertex coloring is a concept in graph theory that refers to assigning colors to the vertices of a graph in such a way that no two adjacent vertices have the same color.
- Formally, the vertex coloring of a graph is an assignment of colors. We usually represent the colors by numbers.
- Additionally, we assign the colors to the vertices such that two adjacent vertices are always filled with different colors.
- The minimum number of colors we require to color the vertices of a graph is known as the chromatic number.
- Generally, this number is always greater than one.
- However, the chromatic number is equal to one if and only if the graph contains a single vertex.
- The concept of vertex coloring is used in many areas of computer science and mathematics.
- Vertex coloring is a way of assigning colors to the vertices of a graph such that no two adjacent verties share the same color.

Date

Example:-
consider the following simple graph G.



steps to color the Graph:-

1) Initialize colors:- Start with a set of colors. three colors:- Red (R), Green (G) and Blue (B).

2) Choose vertex:-
start coloring from any vertex. Let's choose vertex A.

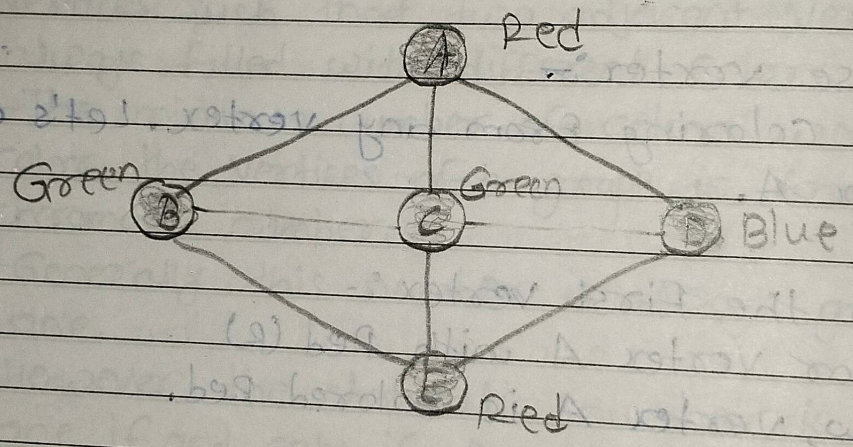
3) Color the First vertex:-
- color Vertex A with Red (R)
- Now, vertex A is colored Red.

4) Move to Adjacent vertices:-
- Move to an adjacent vertex of A, say vertex B.
- since, B is adjacent to A, it cannot be Red. Color B with Green (G).

5) Continue coloring:-
- move to another adjacent vertex of A, say vertex C.
- C is adjacent to A, so it cannot be Red. Color C with Green (G) since its not adjacent to B.

6) NExt vertex:-
- Move to the next adjacent vertex of A, say vertex D.
- D is adjacent to A, so it cannot be Red.
- since D is also adjacent to C (which is Green), we need a different color.
- Color D with Blue (B).

7) color the Remaining vertex:-
- Move to vertex E.
- E is adjacent to B (Green) and D (Blue), so, we can color E with Red (R) as it is not adjacent to any Red vertex.



- Check that no two adjacent vertices share the same color:
1) A(R) is adjacent to B(G), C(G) and D(B), all different colors.
2) B(G) is adjacent to A(R) and E(R), both different colors.
3) C(G) is adjacent to A(R) and D(B), both different colors.
4) D(B) is adjacent to A(R) and E(R), both different colors.

5) E(F) is adjacent to B(G) and D(B), both different colors.

Application of vertex coloring :-
1) Register Allocation in compilers
2) Frequency Assignment
3) Map coloring
4) Timetabling
5) Network Design
6) circuit design

Imp
Que- Explain and Solve max flow ~~and~~ min-cut theorem for a Network Problem.

⇒

max flow - min - cut theorem :-
- The ~~maximum~~ max-flow, min-cut theorem states that the maximum flow from a given source to a given sink in a network is equal to the minimum sum of a cut that separates the source from the sink.

• Flow :-
- For simplicity, flow can be imagined as a physical flow of a fluid through the network that moves from source to sink through the directed edges.
- Each edge will have a flow that cannot be greater than the edge's capacity.
- Think of it as water flowing through a pipe, where the flow cannot be greater the pipe's capacity.

• Cut :-
An s-t cut is partitioning the graph into two disjoint subsets, with the source in one part and sink in other.
- A cut will only be valid if it completely separates the source from the sink such as there is no path connecting the two.
- Alternatively, a cut is a set of edges whose removal divides the network into two halves x and y, where $s \in x$ and $t \in y$.
- moreover, a cut has a capacity that is equal to the sum of the capacities of the edges in the cut.

• Goal :-
The goal is to find the minimum capacity cut that will dictate the maximum flow achievable in a flow network.

- It states that in a flow network, the maximum amount of flow that can be sent from the source to the sink is equal to the capacity of the smallest(minimum) cut that separates the source from the sink.

• key concepts :-

1) Flow Network :- A directed graph where each edge has a capacity and each edge receives a flow. The amount of flow on an edge cannot exceed the capacity of the edge.

2] source (s) :- The starting node where the Flow originates.

3] sink (t) : The ending node where the Flow is consumed.

4] Flow (f) :- The amount of Flow passing through an edge. It must satisfy Flow conservation at each node (except for source and sink).

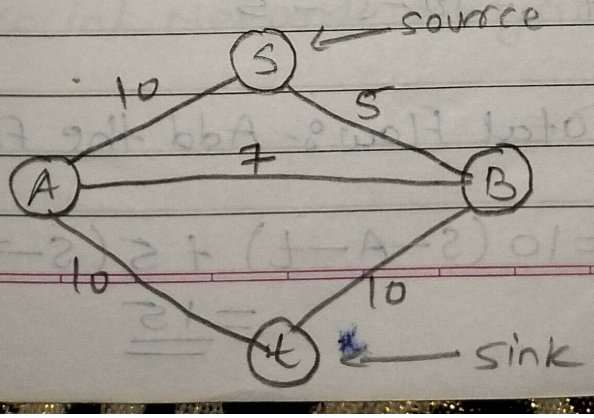
5] capacity (c) :- The maximum amount of Flow that an edge can handle.

6] cut :- A Partition of the vertices of the graph into two disjoint subsets such that one contains the source and the other contains the sink.

7] min cut :- The cut with the smallest capacity that separates the source from the sink.

- Theorem statement :-
The max Flow - min cut theorem states that :-
maximum flow = minimum cut capacity.

Example :-
consider a simple Flow network :-



- Nodes : S (source), t (sink), A, B
- Edge capacities : S - A (10), S - B (5), A - B (7), A - t (10), B - t (10)

steps :-

1) Construct Residual Graph :- Start with the original capacities. The residual capacity of an edge is the original capacity minus the current Flow.

2) Find Augmenting Paths :- use the Ford-Fulkerson method to find augmenting Paths from S to t.

- First Augmenting Path :-
  - Path :- S → A → t
  - Flow :- min (10, 10) = 10
  - update capacities :-
    a) S → A :- 10 - 10 = 0
    b) A → t :- 10 - 10 = 0

- Second Augmenting Path :-
  - Path :- S → B → t
  - Flow :- min (5, 10) = 5
  - update capacities :-
    a) S → B :- 5 - 5 = 0
    b) B → t :- 10 - 5 = 5

3) Calculate Total Flow :- Add the Flows from each Path.

- Total Flow = 10 (S - A - t) + 5 (S → B → t)
  = 15

4) Identify minimum cut :-

- The remaining capacities in the residual graph are :-
  - $S \rightarrow A$:- 0 (Full)
  - $S \rightarrow B$:- 0 (Full)
  - $A \rightarrow B$:- 7 (not used)
  - $A \rightarrow t$:- 0 (Full)
  - $B \rightarrow t$:- 5 (Partial)

- Cut edges to minimize the flow from source to sink :-
  - one possible cut is $\{S\}, \{A, B, t\}$ with cut capacity $= 10 (S \rightarrow A) + 5 (S \rightarrow B)$
    $= \underline{15}$

Therefore, the capacity of the minimum cut is 15, which matches the total maximum flow calculated, confirming the max flow-min cut Theorem.

$\therefore$ Maximum flow = minimum cut capacity

$15 = 15$

Applications of max flow - min cut :-

1) Network Design and Analysis
2) Image Segmentation in computer vision
3) Bipartite Matching
4) Circulation and Network Flow
5) social Network Analysis

Page No.
Date

Que Explain Probabilistic models in graph.

⇒

Probabilistic models :-

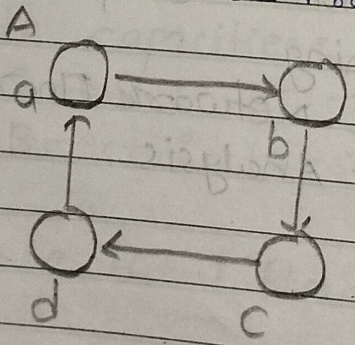
- Probabilistic models in graph data structures refer to approaches where probability theory is utilized to model uncertain or probabilistic relationships between nodes or edges in a graph.
- These models are particularly useful in scenarios where the underlying data is noisy, incomplete, or uncertain, such as in social networks, biological networks and sensor networks.

Types of Probabilistic models :-

1) Graphical model
2) Generative model
3) Discriminative model

1) Graphical model :-

- Graphical models are a powerful tool for understanding and reasoning about complex probabilistic relationships.
- They provide a visual and intuitive way to represent the dependencies between random variables, making it easier to model, analyze and draw inferences from data.



Page No.
Date

1] Directed Graphical models (Bayesian Network)

- Directed Acyclic Graphs :-
  - Bayesian networks are directed acyclic graphs (DAGs) that represent conditional dependencies between random variables.
- Probabilistic Inference :-
  - Bayesian networks allow for efficient probabilistic inference, where we can reason about the state of unobserved variables given the observed ones.
- Parameter Learning :-
  The parameters of a Bayesian network, such as conditional probability distributions, can be learned from data using ~~steel~~ statistical techniques.

2] Undirected Graphical models (Markov Random Fields) :-

- Markov Random Fields (MRFs) are a type of undirected graphical model that represent the probabilistic relationships between variables using an undirected graph.
- In an MRF, the nodes represent random variables and the edges represent the statistical dependencies between them, without indicating the direction of the relationship.
- MRFs are well-suited for modeling spatial and contextual data, where the value of a variable depends on the values of its neighboring variables in the graph.

Page No.
Date

- Application of Graphical models :-

1) Probabilistic Reasoning
2) Anomaly Detection
3) Decision Support
4) Causal Inference