Graphs

Madhavan Mukund

https://www.cmi.ac.in/~madhavan

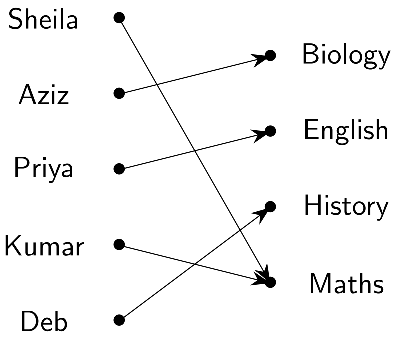
Programming, Data Structures and Algorithms using Python

Week 4

Visualizing relations as graphs

- Teachers and courses
  - T, set of teachers in a college
    C, set of courses being offered
  - $A \subseteq T \times C$ describes the allocation of teachers to courses
  - $A = \{(t, c) \mid (t, c) \in T \times C, t \text{ teaches } c\}$

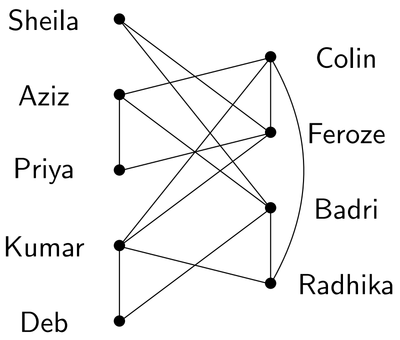
Teachers and courses



Visualizing relations as graphs

- Teachers and courses
  - T, set of teachers in a college
    C, set of courses being offered
  - $A \subseteq T \times C$ describes the allocation of teachers to courses
  - $A = \{(t,c) \mid (t,c) \in T \times C, t \text{ teaches } c\}$
- Friendships
  - P, a set of students
  - $F \subseteq P \times P$ describes which pairs of students are friends
  - $F = \{(p,q) \mid p,q \in P, p \neq q, p \text{ is a friend of } q\}$
  - $(p,q) \in F$ iff $(q,p) \in F$

Friendship



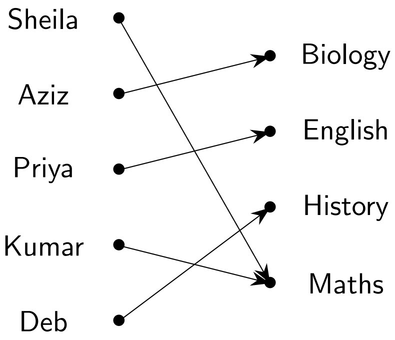
Graphs

- Graph: $G = (V, E)$
  - V is a set of vertices or nodes
    - One vertex, many vertices
  - E is a set of edges
  - $E \subseteq V \times V$ — binary relation

Graphs

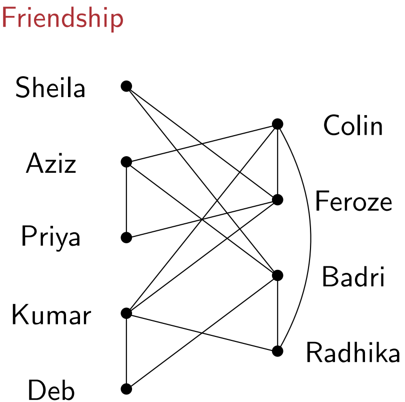
- Graph: $G = (V, E)$
  - V is a set of vertices or nodes
    - One vertex, many vertices
  - E is a set of edges
  - $E \subseteq V \times V$ — binary relation
- Directed graph
  - $(v, v') \in E$ does not imply $(v', v) \in E$
  - The teacher-course graph is directed

Teachers and courses



Graphs

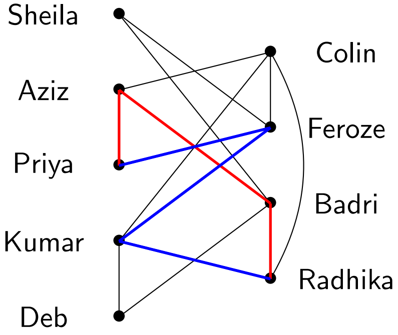
- Graph: $G = (V, E)$
  - V is a set of vertices or nodes
    - One vertex, many vertices
  - E is a set of edges
  - $E \subseteq V \times V$ — binary relation
- Directed graph
  - $(v, v') \in E$ does not imply $(v', v) \in E$
  - The teacher-course graph is directed
- Undirected graph
  - $(v, v') \in E$ iff $(v', v) \in E$
  - Effectively (v, v'), (v', v) are the same edge
  - Friendship graph is undirected



Paths

- A path is a sequence of vertices $v_1, v_2, \ldots, v_k$ connected by edges
  - For $1 \leq i < k$, $(v_i, v_{i+1}) \in E$

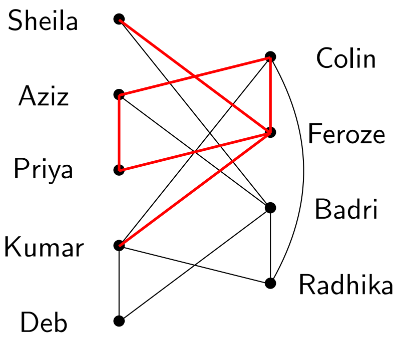
Friendship graph



Paths

- A path is a sequence of vertices $v_1, v_2, \ldots, v_k$ connected by edges
  - For $1 \leq i < k$, $(v_i, v_{i+1}) \in E$
- Normally, a path does not visit a vertex twice
- A sequence that re-visits a vertex is usually called a walk
  - Kumar — Feroze — Colin — Aziz — Priya — Feroze — Sheila

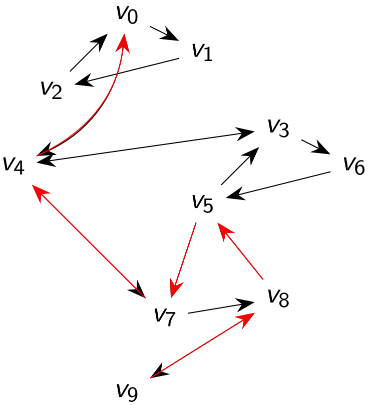
Friendship graph



Reachability

- Paths in directed graphs
- How can I fly from Madurai to Delhi?
  - Find a path from v_9 to v_0
- Vertex v is reachable from vertex u if there is a path from u to v

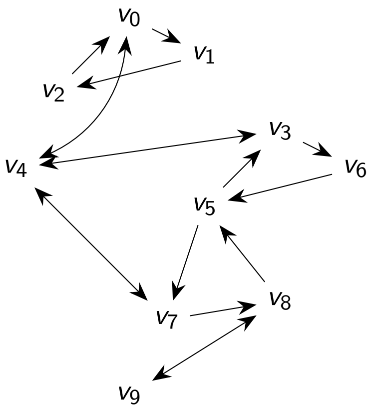
Airline routes



Reachability

- Paths in directed graphs
- How can I fly from Madurai to Delhi?
  - Find a path from v_9 to v_0
- Vertex v is reachable from vertex u if there is a path from u to v
- Typical questions
  - Is v reachable from u?
  - What is the shortest path from u to v?
  - What are the vertices reachable from u?
  - Is the graph connected? Are all vertices reachable from each other?

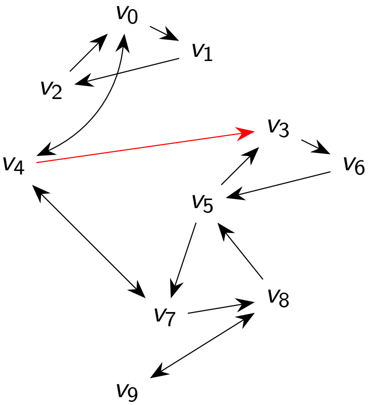
Airline routes



Reachability

- Paths in directed graphs
- How can I fly from Madurai to Delhi?
  - Find a path from v_9 to v_0
- Vertex v is reachable from vertex u if there is a path from u to v
- Typical questions
  - Is v reachable from u?
  - What is the shortest path from u to v?
  - What are the vertices reachable from u?
  - Is the graph connected? Are all vertices reachable from each other?

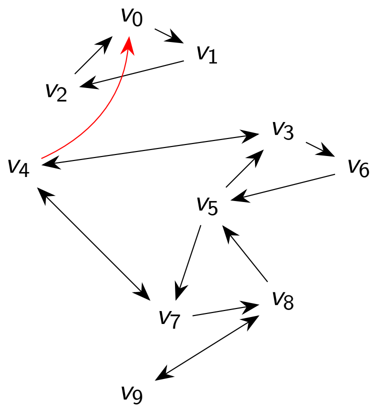
Airline routes



Reachability

- Paths in directed graphs
- How can I fly from Madurai to Delhi?
  - Find a path from v_9 to v_0
- Vertex v is reachable from vertex u if there is a path from u to v
- Typical questions
  - Is v reachable from u?
  - What is the shortest path from u to v?
  - What are the vertices reachable from u?
  - Is the graph connected? Are all vertices reachable from each other?

Airline routes



Map colouring

- Assign each state a colour
- States that share a border should be coloured differently
- How many colours do we need?

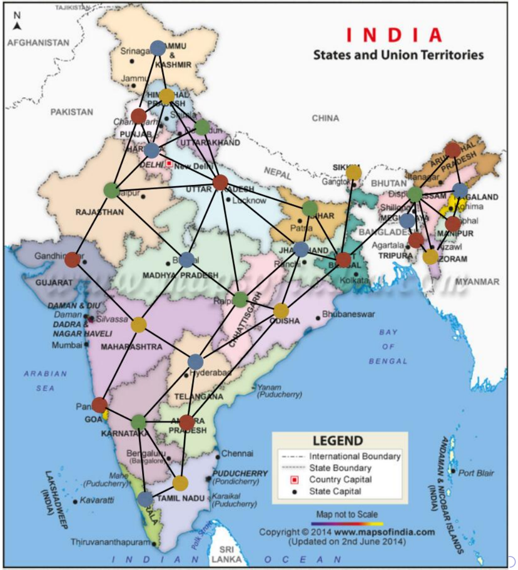
- Assign each state a colour
- States that share a border should be coloured differently
- How many colours do we need?
- Create a graph
  - Each state is a vertex

Map colouring

- Assign each state a colour
- States that share a border should be coloured differently
- How many colours do we need?
- Create a graph
  - Each state is a vertex
  - Connect states that share a border

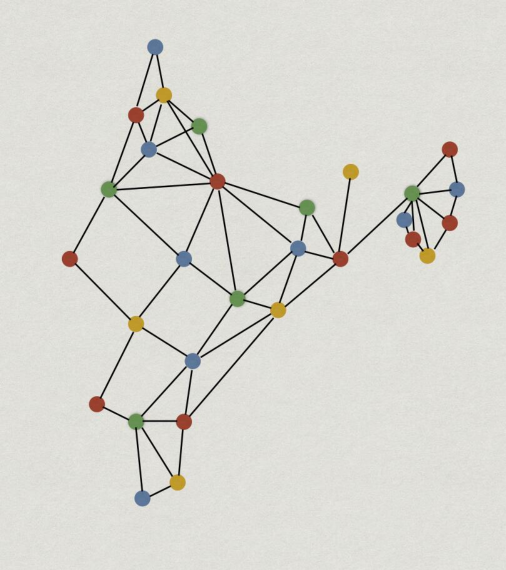
Map colouring

- Assign each state a colour
- States that share a border should be coloured differently
- How many colours do we need?
- Create a graph
  - Each state is a vertex
  - Connect states that share a border
- Assign colours to nodes so that endpoints of an edge have different colours



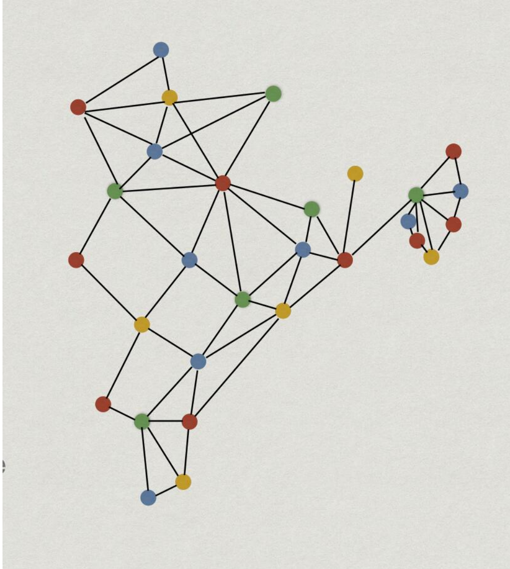
Map colouring

- Assign each state a colour
- States that share a border should be coloured differently
- How many colours do we need?
- Create a graph
  - Each state is a vertex
  - Connect states that share a border
- Assign colours to nodes so that endpoints of an edge have different colours
- Only need the underlying graph



Map colouring

- Assign each state a colour
- States that share a border should be coloured differently
- How many colours do we need?
- Create a graph
  - Each state is a vertex
  - Connect states that share a border
- Assign colours to nodes so that endpoints of an edge have different colours
- Only need the underlying graph
- Abstraction: if we distort the graph, problem is unchanged



Graph colouring

- Graph $G = (V, E)$, set of colours C
- Colouring is a function $c : V \to C$ such that $(u, v) \in E \Rightarrow c(u) \neq c(v)$
- Given $G = (V, E)$, what is the smallest set of colours need to colour G

Graph colouring

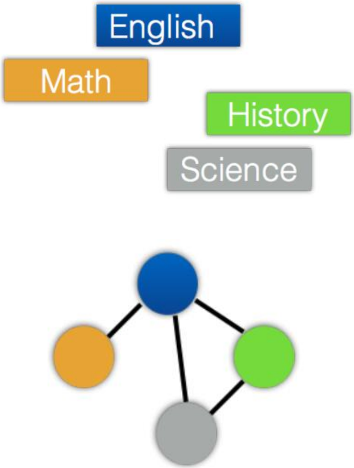
- Graph $G = (V, E)$, set of colours C
- Colouring is a function $c : V \to C$ such that $(u, v) \in E \Rightarrow c(u) \neq c(v)$
- Given $G = (V, E)$, what is the smallest set of colours need to colour G
  - Four Colour Theorem For planar graphs derived from geographical maps, 4 colours suffice

Graph colouring

- Graph $G = (V, E)$, set of colours C
- Colouring is a function $c : V \to C$ such that $(u, v) \in E \Rightarrow c(u) \neq c(v)$
- Given $G = (V, E)$, what is the smallest set of colours need to colour G
  - Four Colour Theorem For planar graphs derived from geographical maps, 4 colours suffice
  - Not all graphs are planar. General case? Why do we care?

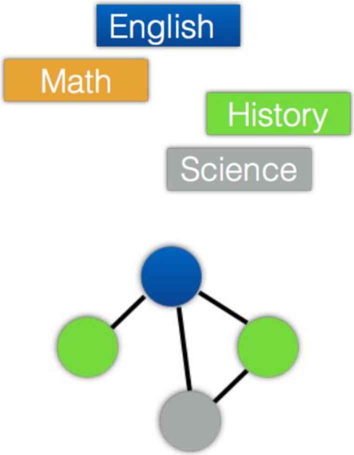
Graph colouring

- Graph $G = (V, E)$, set of colours C
- Colouring is a function $c : V \to C$ such that $(u, v) \in E \Rightarrow c(u) \neq c(v)$
- Given $G = (V, E)$, what is the smallest set of colours need to colour G
  - Four Colour Theorem For planar graphs derived from geographical maps, 4 colours suffice
  - Not all graphs are planar. General case? Why do we care?
- How many classrooms do we need?
  - Courses and timetable slots, edges represent overlapping slots
  - Colours are classrooms



Graph colouring

- Graph $G = (V, E)$, set of colours C
- Colouring is a function $c : V \to C$ such that $(u, v) \in E \Rightarrow c(u) \neq c(v)$
- Given $G = (V, E)$, what is the smallest set of colours need to colour G
  - Four Colour Theorem For planar graphs derived from geographical maps, 4 colours suffice
  - Not all graphs are planar. General case? Why do we care?
- How many classrooms do we need?
  - Courses and timetable slots, edges represent overlapping slots
  - Colours are classrooms

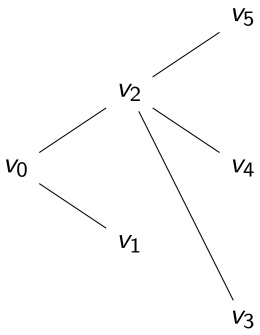


Vertex cover

- A hotel wants to install security cameras
  - All corridors are straight lines
  - Camera can monitor all corridors that meet at an intersection
- Minimum number of cameras needed?

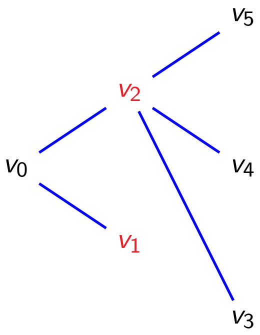
Vertex cover

- A hotel wants to install security cameras
  - All corridors are straight lines
  - Camera can monitor all corridors that meet at an intersection
- Minimum number of cameras needed?
- Represent the floor plan as a graph
  - V — intersections of corridors
  - E — corridor segments connecting intersections



Vertex cover

- A hotel wants to install security cameras
  - All corridors are straight lines
  - Camera can monitor all corridors that meet at an intersection
- Minimum number of cameras needed?
- Represent the floor plan as a graph
  - V — intersections of corridors
  - E — corridor segments connecting intersections
- Vertex cover
  - Marking v covers all edges from v
  - Mark smallest subset of V to cover all edges

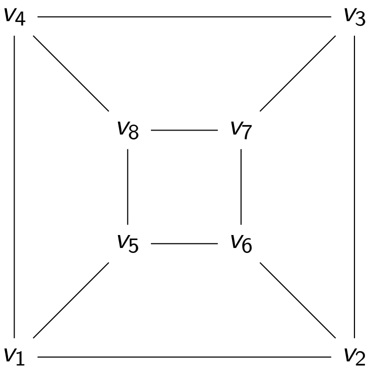


Independent set

- A dance school puts up group dances
  - Each dance has a set of dancers
  - Sets of dancers may overlap across dances
- Organizing a cultural programme
  - Each dancer performs at most once
  - Maximum number of dances possible?

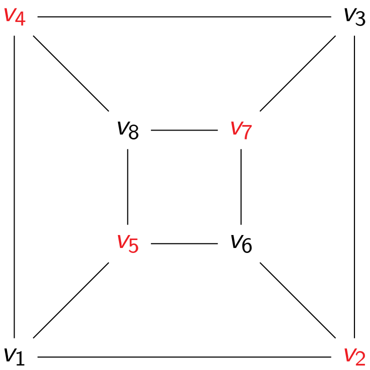
Independent set

- A dance school puts up group dances
  - Each dance has a set of dancers
  - Sets of dancers may overlap across dances
- Organizing a cultural programme
  - Each dancer performs at most once
  - Maximum number of dances possible?
- Represent the dances as a graph
  - V — dances
  - E — sets of dancers overlap



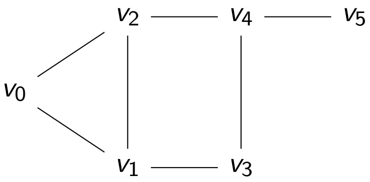
Independent set

- A dance school puts up group dances
  - Each dance has a set of dancers
  - Sets of dancers may overlap across dances
- Organizing a cultural programme
  - Each dancer performs at most once
  - Maximum number of dances possible?
- Represent the dances as a graph
  - V — dances
  - E — sets of dancers overlap
- Independent set
  - Subset of vertices such that no two are connected by an edge



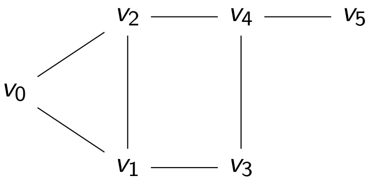
Matching

- Class project can be done by one or two people
  - If two people, they must be friends
- Assume we have a graph describing friendships



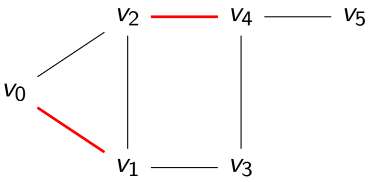
Matching

- Class project can be done by one or two people
  - If two people, they must be friends
- Assume we have a graph describing friendships
- Find a good allocation of groups
- Matching
  - $G = (V, E)$, an undirected graph
  - A matching is a subset $M \subseteq E$ of mutually disjoint edges



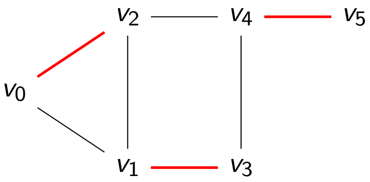
Matching

- Class project can be done by one or two people
  - If two people, they must be friends
- Assume we have a graph describing friendships
- Find a good allocation of groups
- Matching
  - $G = (V, E)$, an undirected graph
  - A matching is a subset $M \subseteq E$ of mutually disjoint edges
- Find a maximal matching in G



Matching

- Class project can be done by one or two people
  - If two people, they must be friends
- Assume we have a graph describing friendships
- Find a good allocation of groups
- Matching
  - $G = (V, E)$, an undirected graph
  - A matching is a subset $M \subseteq E$ of mutually disjoint edges
- Find a maximal matching in G
- Is there a perfect matching, covering all vertices?



Summary

- A graph represents relationships between entities
  - Entities are vertices/nodes
  - Relationships are edges
- A graph may be directed or undirected
  - A is a parent of B — directed
  - A is a friend of B — undirected

Summary

- A graph represents relationships between entities
  - Entities are vertices/nodes
  - Relationships are edges
- A graph may be directed or undirected
  - A is a parent of B — directed
  - A is a friend of B — undirected
- Paths are sequences of connected edges
- Reachability: is there a path from u to v?

Summary

- A graph represents relationships between entities
  - Entities are vertices/nodes
  - Relationships are edges
- A graph may be directed or undirected
  - A is a parent of B — directed
  - A is a friend of B — undirected
- Paths are sequences of connected edges
- Reachability: is there a path from u to v?
- Graphs are useful abstract representations for a wide range of problems
- Reachability and connectedness are not the only interesting problems we can solve on graphs
  - Graph colouring
  - Vertex cover
  - Independent set
  - Matching
  - ...