More on Graphs

Madhavan Mukund

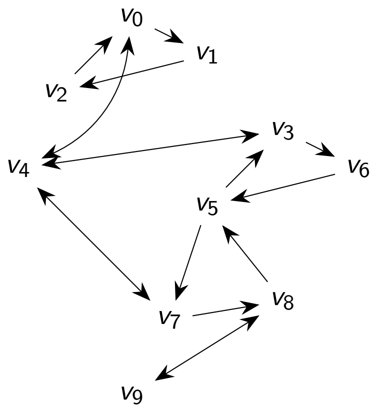
https://www.cmi.ac.in/~madhavan

Mathematics for Data Science 1
Week 10

Graphs

- Graph $G = (V, E)$
  - V — set of vertices
  - $E \subseteq V \times V$ — set of edges

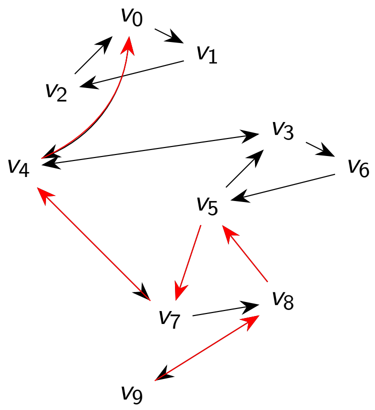
Airline routes



Graphs

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

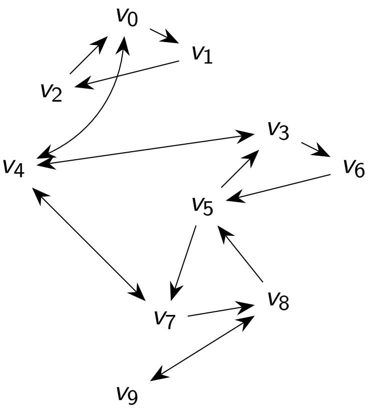
Airline routes



Graphs

- Graph $G = (V, E)$
  - V — set of vertices
  - $E \subseteq V \times V$ — set of edges
- A path is a sequence of vertices $v_1, v_2, \ldots, v_k$ connected by edges
  - For $1 \leq i < k$, $(v_i, v_{i+1}) \in E$
- Vertex v is reachable from vertex u if there is a path from u to v

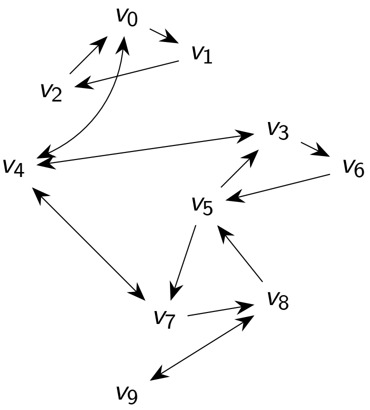
Airline routes



Graphs

- Graph $G = (V, E)$
  - V — set of vertices
  - $E \subseteq V \times V$ — set of edges
- A path is a sequence of vertices $v_1, v_2, \ldots, v_k$ connected by edges
  - For $1 \leq i < k$, $(v_i, v_{i+1}) \in E$
- Vertex v is reachable from vertex u if there is a path from u to v
- What more can we do with graphs?

Airline routes



- Assign each state a colour
- States that share a border should be coloured differently

Map colouring

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

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

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

Map colouring

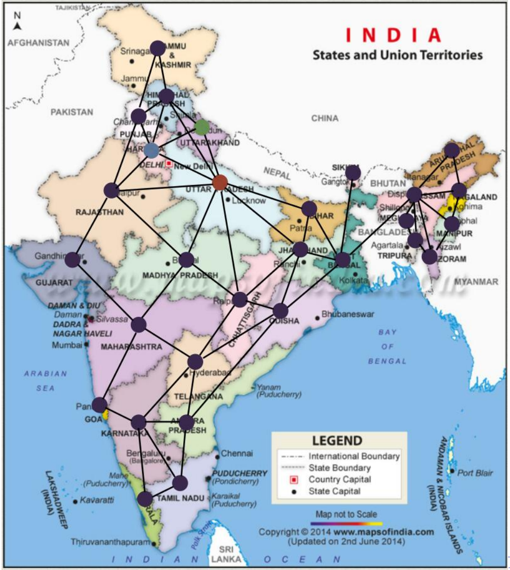
- Assign each state a colour
- States that share a border should be coloured differently
- How many colours do we need?
- Create a graph
  - Each state is a vertex
  - Connect states that share a border

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

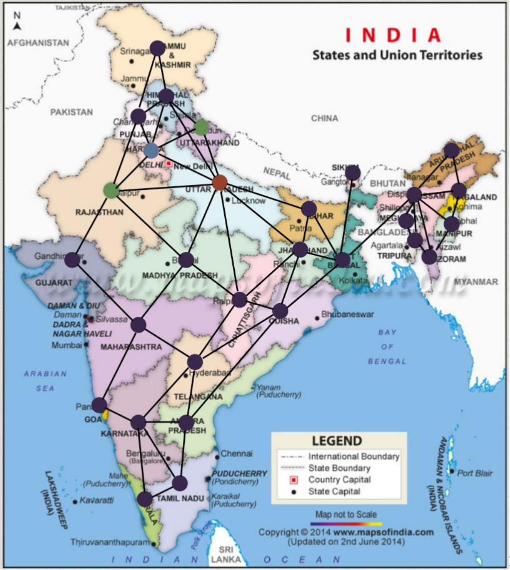
Map colouring

- 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



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

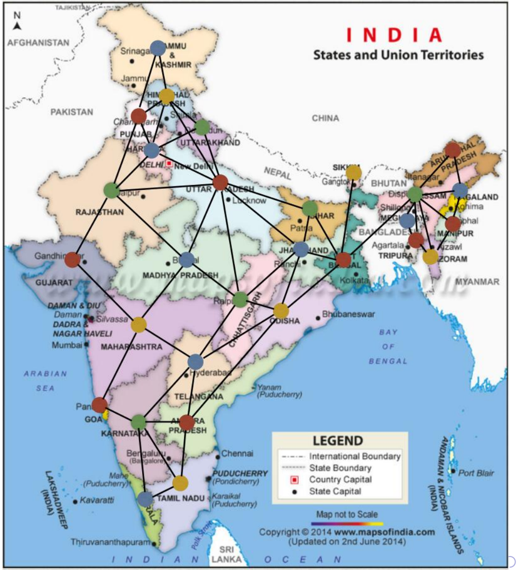
- Assign each state a colour
- States that share a border should be coloured differently
- How many colours do we need?
- Create a graph
  - Each state is a vertex
  - Connect states that share a border
- Assign colours to nodes so that endpoints of an edge have different colours

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

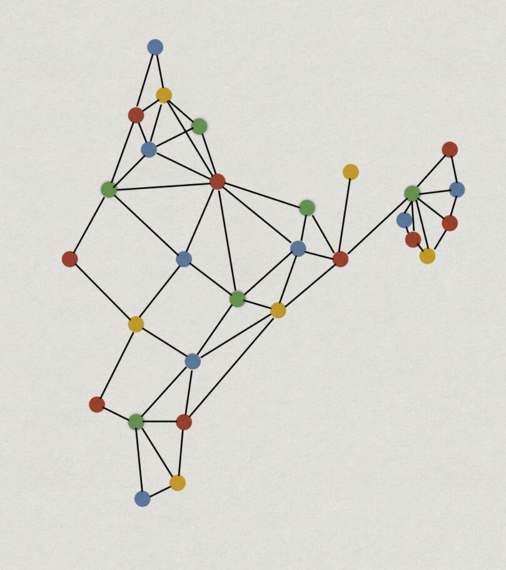
Map colouring

- 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



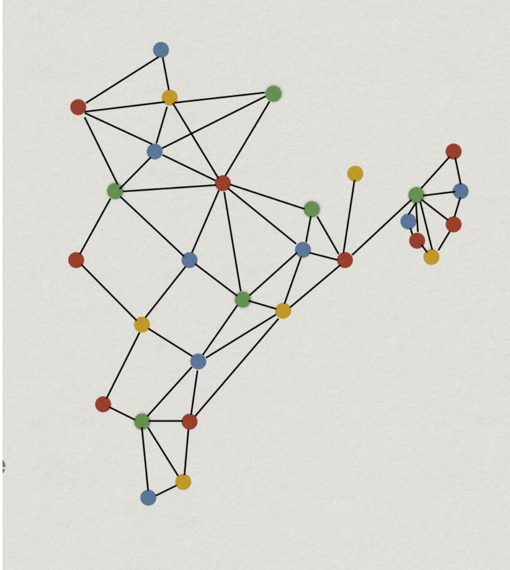
Map colouring

- 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

Graph colouring

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

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 graphs derived from geographical maps, 4 colours suffice

Graph colouring

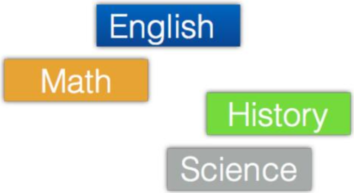
- Graph $G = (V, E)$, set of colours C
- Colouring is a function $c : V \to C$ such that $(u, v) \in E \Rightarrow c(u) \neq c(v)$
- Given $G = (V, E)$, what is the smallest set of colours need to colour G
  - Four Colour Theorem For graphs derived from geographical maps, 4 colours suffice
  - Not all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

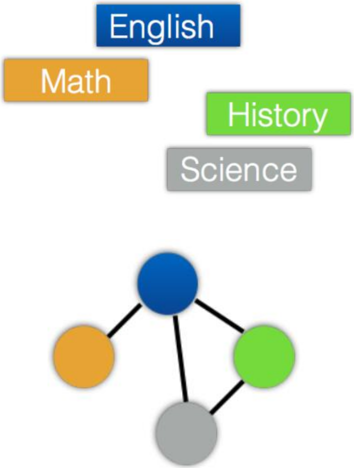
Graph colouring

- 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 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



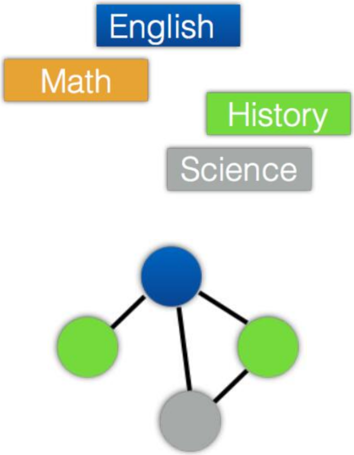
Graph colouring

- 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 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
  - Graph: Edges are overlaps in slots



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 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
  - Graph: Edges are overlaps in slots
  - Colours are classrooms



Vertex cover

- A hotel wants to install security cameras
  - All corridors are straight lines
  - Camera at the intersection of corridors can monitor all those corridor.

Vertex cover

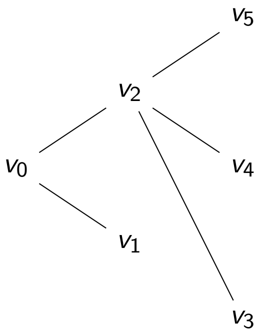
- A hotel wants to install security cameras
  - All corridors are straight lines
  - Camera at the intersection of corridors can monitor all those corridor.
- Minimum number of cameras needed?

Vertex cover

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

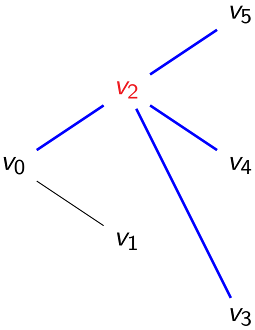
Vertex cover

- A hotel wants to install security cameras
  - All corridors are straight lines
  - Camera at the intersection of corridors can monitor all those corridor.
- 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



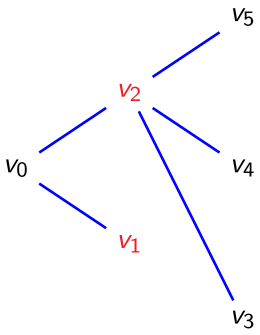
Vertex cover

- A hotel wants to install security cameras
  - All corridors are straight lines
  - Camera at the intersection of corridors can monitor all those corridor.
- 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



Vertex cover

- A hotel wants to install security cameras
  - All corridors are straight lines
  - Camera at the intersection of corridors can monitor all those corridor.
- 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

Independent set

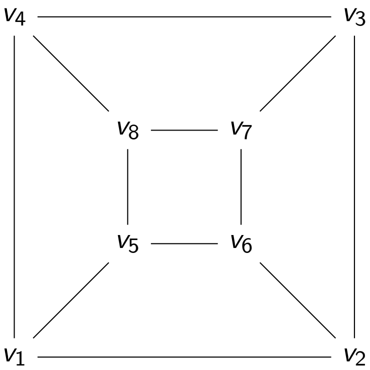
- A dance school puts up group dances
  - Each dance has a set of dancers
  - Sets of dancers may overlap across dances
- Organizing a cultural programme
  - Each dancer performs at most once
  - Maximum number of dances possible?

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

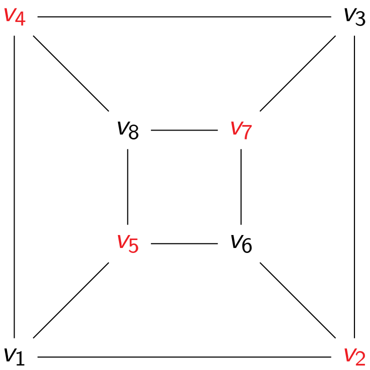
Independent set

- 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



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

Matching

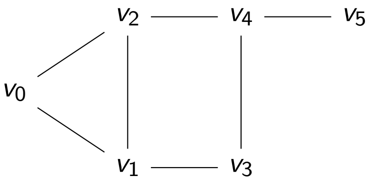
- Class project can be done by one or two people
  - If two people, they must be friends
- Assume we have a graph describing friendships

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

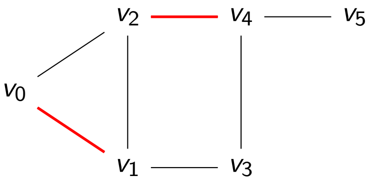
Matching

- 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



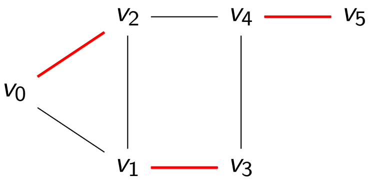
Matching

- 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

- 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
  - ...