Lecture 9

Markets, Mechanisms and Machines

David Evans and Denis Nekipelov

Graphs

- Objects that provide models of relationships

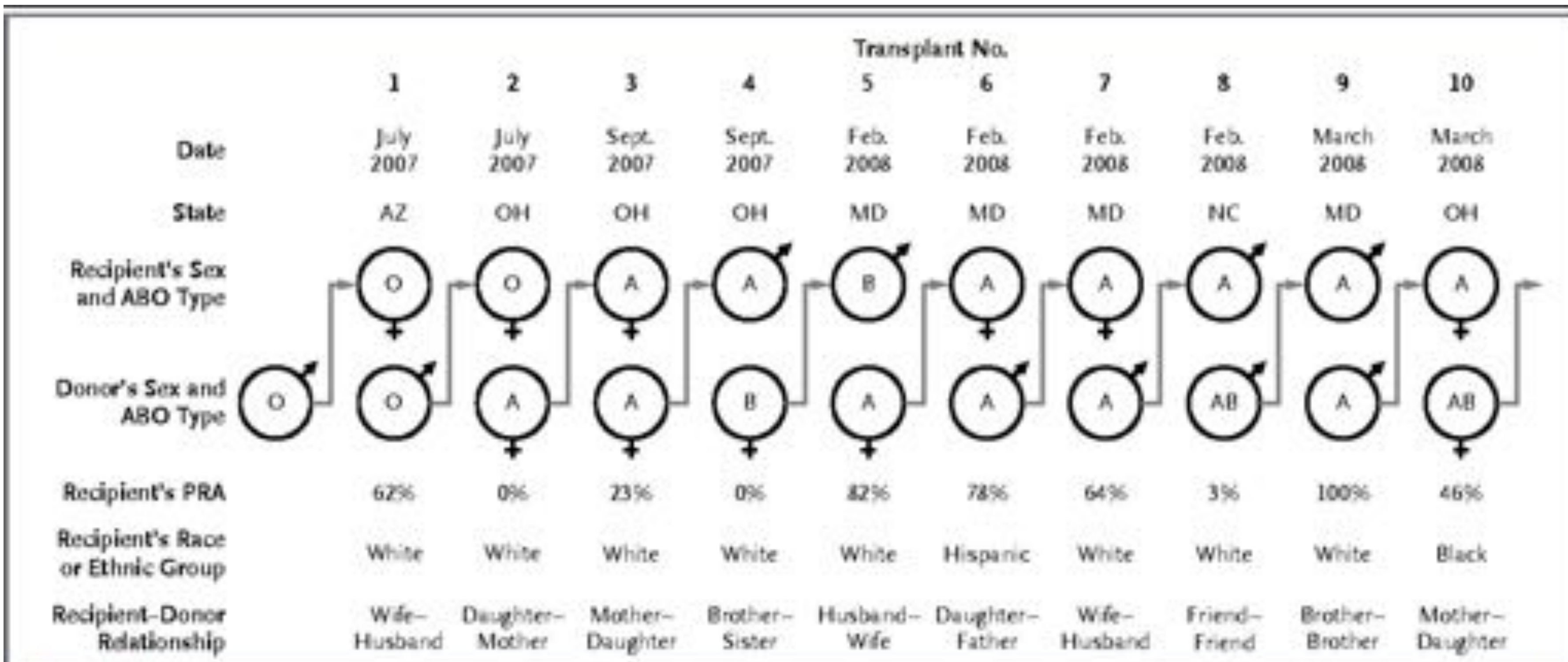
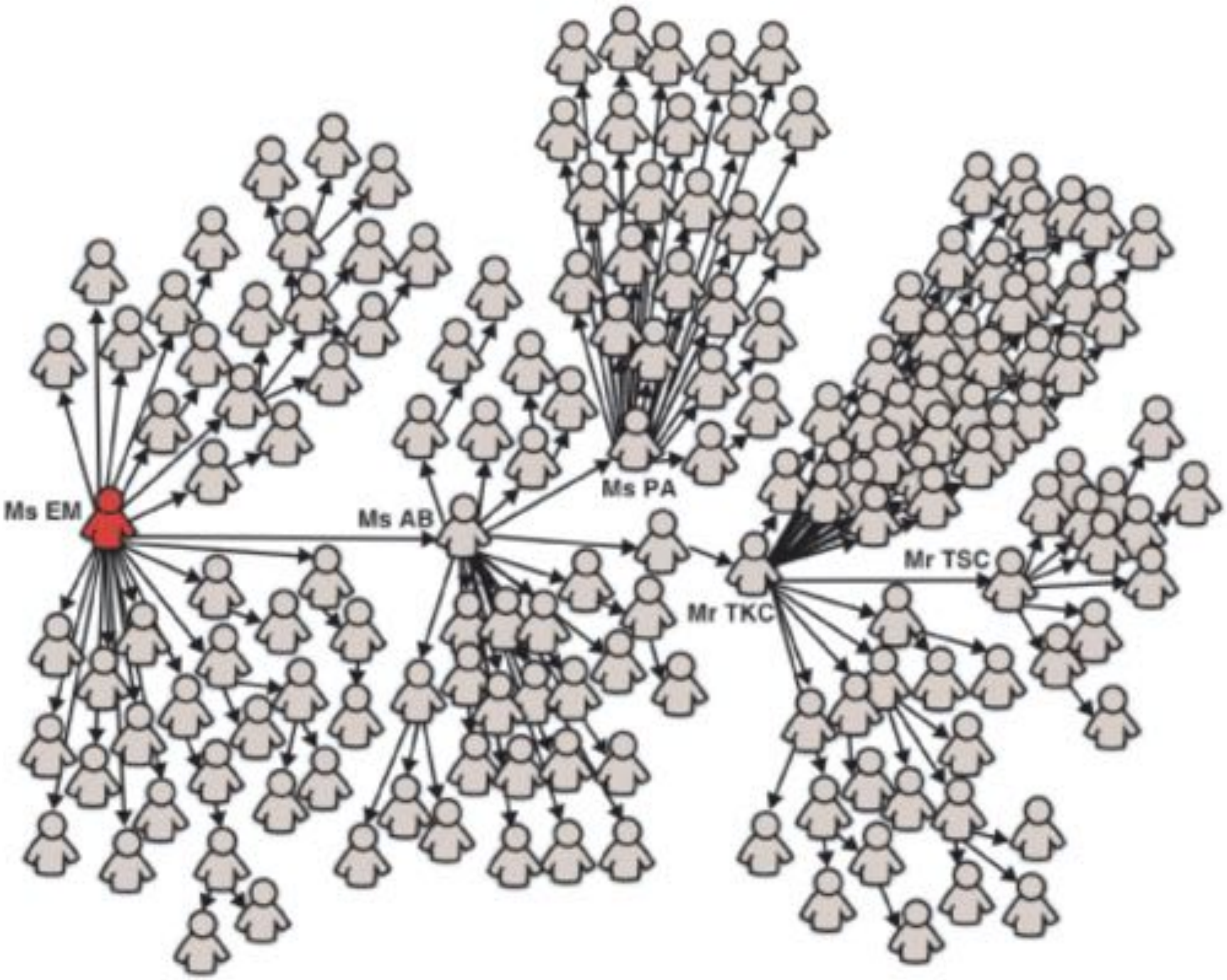


Figure 1. A Nonsimultaneous, Extended, Altruistic-Donor Chain.

In less than 1 year, 10 patients (5 with panel-reactive antibodies [PRAs] >60%) were given a transplant; an 11th transplantation is possible. The initiating donor was an unpaired altruistic donor from Michigan. To date, none of the bridge donors have reneged. The recipient of transplant 6 required desensitization of HLA donor-specific antibodies by T-cell and B-cell flow cytometry. The recipient of transplant 9 had an anti-B antibody titer of 1:8 (as assessed with the use of an antihuman globulin reagent) and required desensitization to blood group.

Graphs

- Objects that provide models of relationships

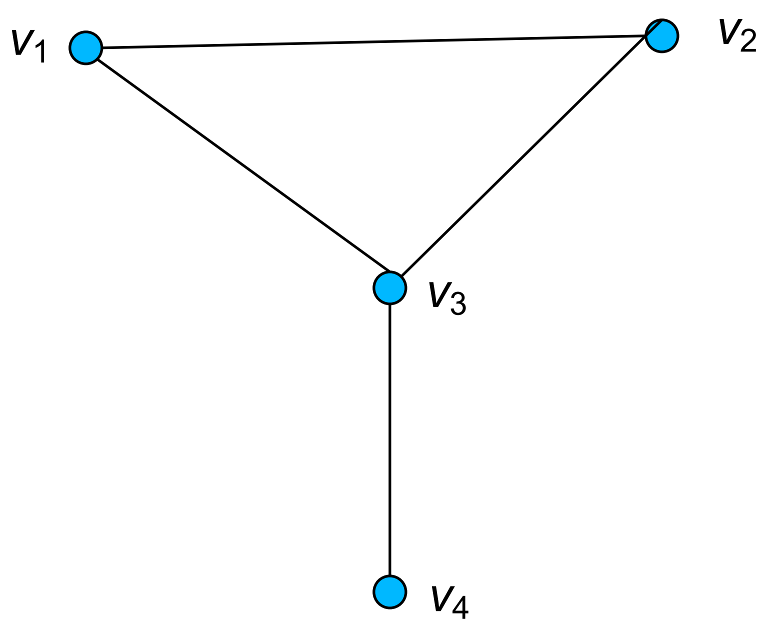


Graphs

- Graph G
  - Finite set V + symmetric irreflexive relation R on V
  - Set E contains all symmetric pairs in R
- Example
  - $V = \{v_1, v_2, v_3, v_4\}$
  - $R = \{(v_1, v_2), (v_1, v_3), (v_2, v_1), (v_2, v_3), (v_3, v_1), (v_3, v_2),$
    $(v_3, v_4), (v_4, v_3)\}$
  - $E = \{\{(v_1, v_2), (v_2, v_1)\}, \{(v_1, v_3), (v_3, v_1)\},$
    $\{(v_2, v_3), (v_3, v_2)\},\{(v_3, v_4), (v_4, v_3)\}\}$

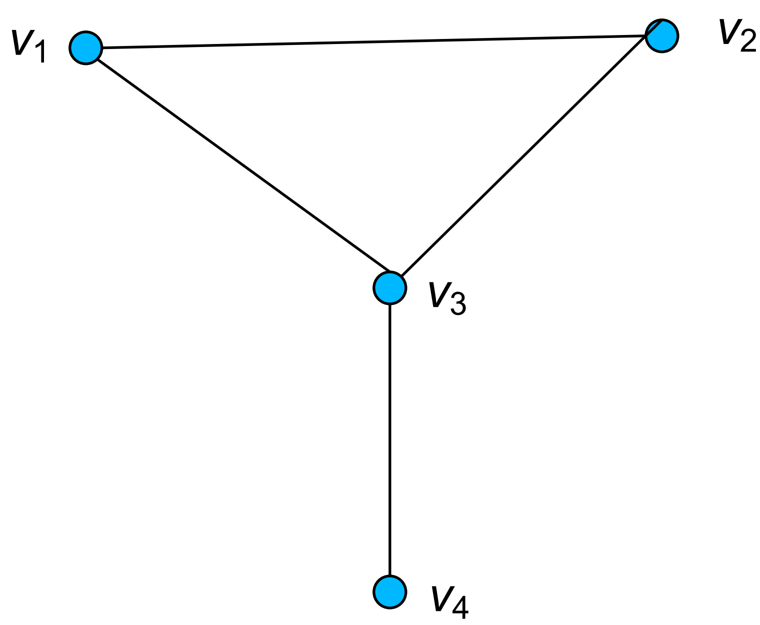
Graphs

- If $e = uv \in E(G)$, edge e joins vertices u and v
- Vertices u and v are adjacent if $uv \in E(G)$ (otherwise they are nonadjacent)
- If $uv \in E(G)$ and $uw \in E(G)$ then uv and uw are adjacent edges



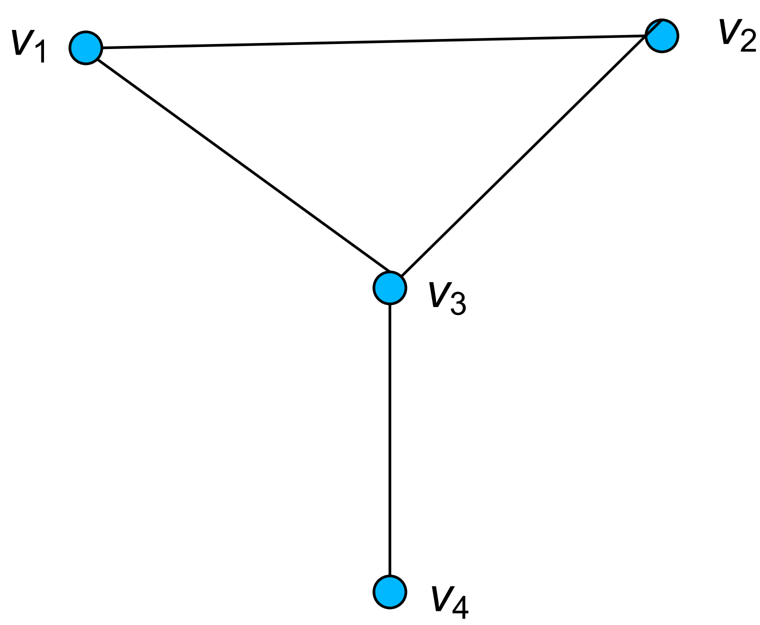
Graphs

- In graph *G, V* is the vertex set
- Set *E* is the set of edges of *G*
- $|V|$is called the order of G and $|E|$ is the size of G
- The diagram of G completely describes it and can itself be called a graph



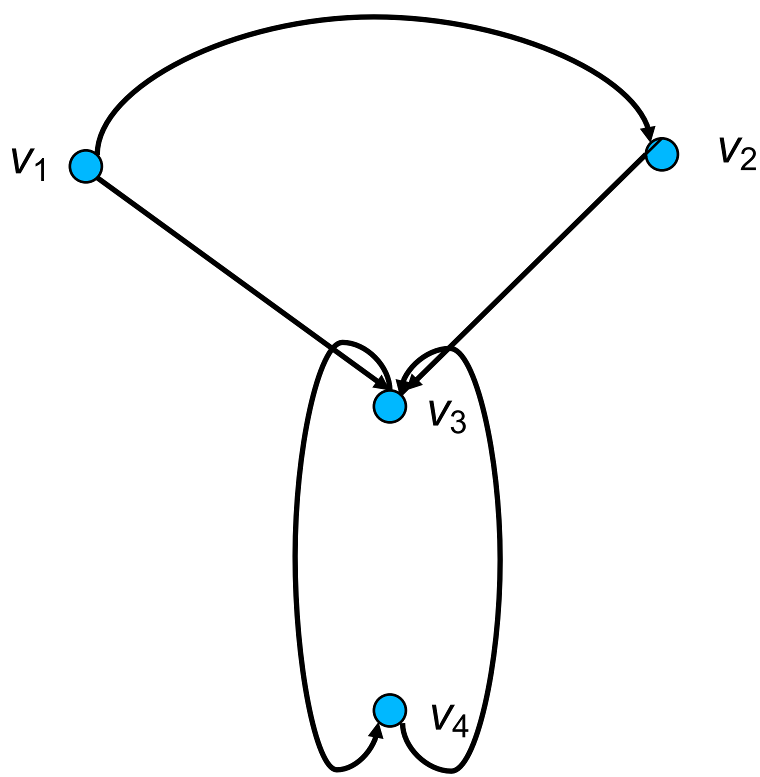
Graphs

- If $e = uv \in E(G)$, edge e joins vertices u and v
- Vertices u and v are adjacent if $uv \in E(G)$ (otherwise they are nonadjacent); u is called incident to v
- If $uv \in E(G)$ and $uw \in E(G)$ then uv and uw are adjacent edges



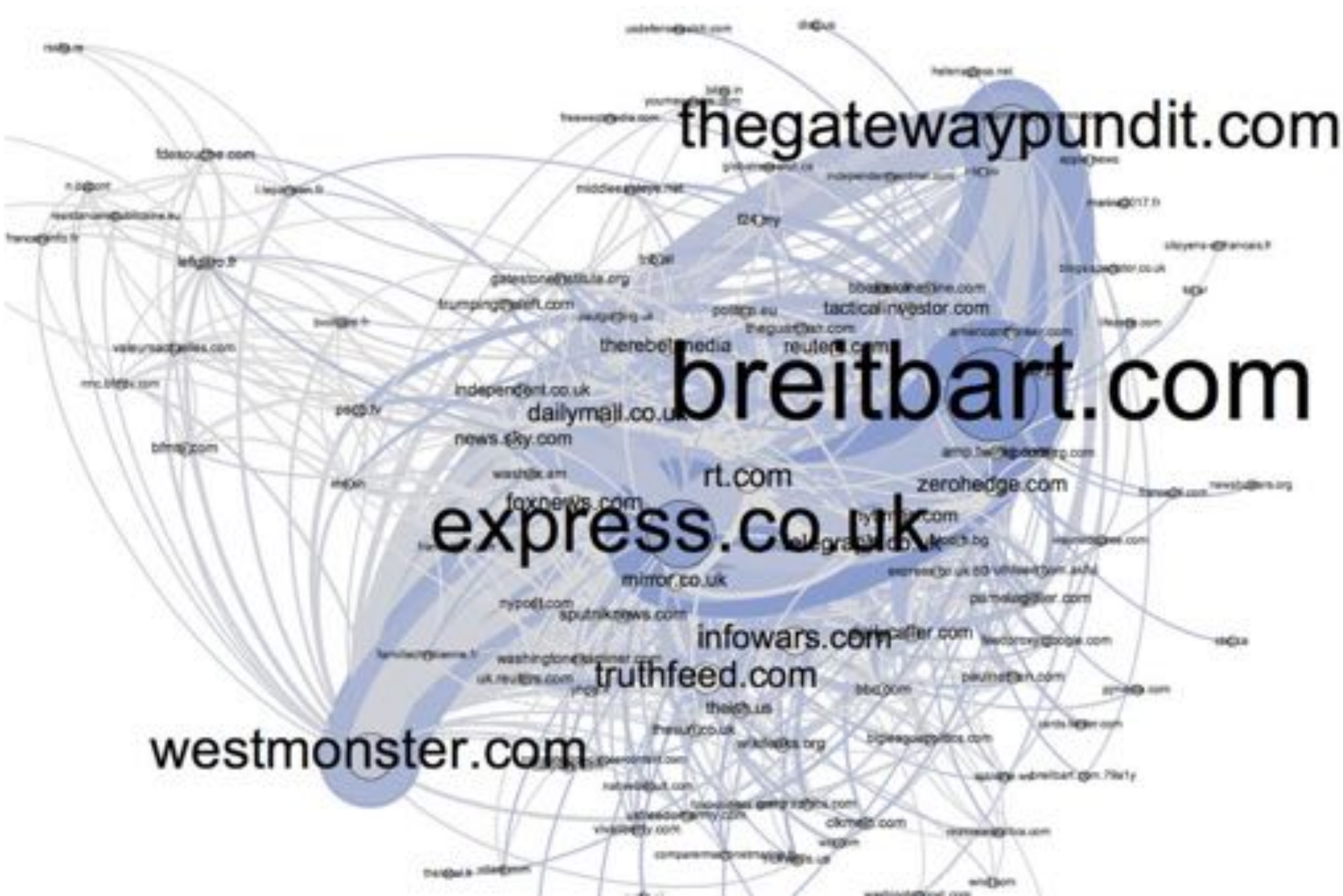
Graphs

- A directed graph *D* is finite set *V* and irreflexive (by not necessarily symmetric) relation *R*
- Each ordered pair in *R* is a directed edge (arc)
- If vertex *v* is connected with *u*, *u* may not be connected with *v*



Networks

- A network is a graph + function that maps edges into real numbers
    - Networks can be directed or undirected



Basic properties of graphs

- Number of edges incident with $v \in V(G)$ is called the degree of v
- A vertex is even or odd if its degree is even or odd
- (p,q) graph is graph with order p and size q

Theorem

For any (p,q) graph G $\sum_{v \in V(G)} \deg(v)=2q$ (sum of degrees of vertices is twice the number of edges)

Theorem

Every graph contains an even number of odd vertices

- If $\deg(v)=r$ for all $v \in V(G)$, G is called r-regular
- G is complete if for all $v, u \in V(G)$, $uv \in E(G)$ (it is then $(p-1)$-regular)

Basic properties of graphs

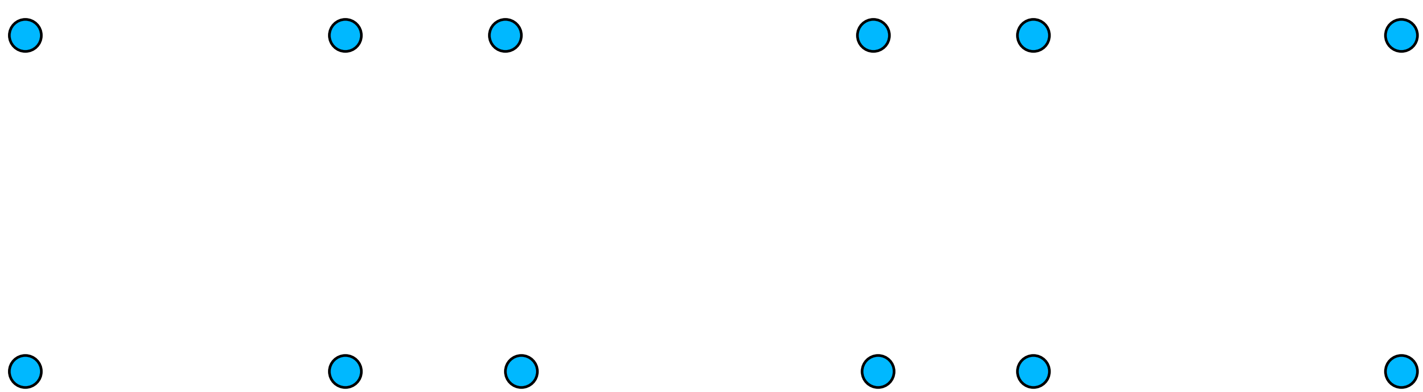
- Recall that isomorphism from G_1 to G_2 is a one-to-one mapping ϕ: $V(G_1) \longmapsto V(G_2)$
  - For any $u, v \in V(G_1)$ s.t. $uv \in E(G_1)$, $\phi(u)\phi(v) \in E(G_2)$
- G_1 to G_2 are isomorphic is isomorphism ϕ exists
  - G_1 can be redrawn to look identical to G_2
- Isomorphism defines an equivalence class of graphs

Theorem

If G_1 and G_2 are isomprhic, then degrees of vertices of G_1 are identical to degrees of vertices of G_2

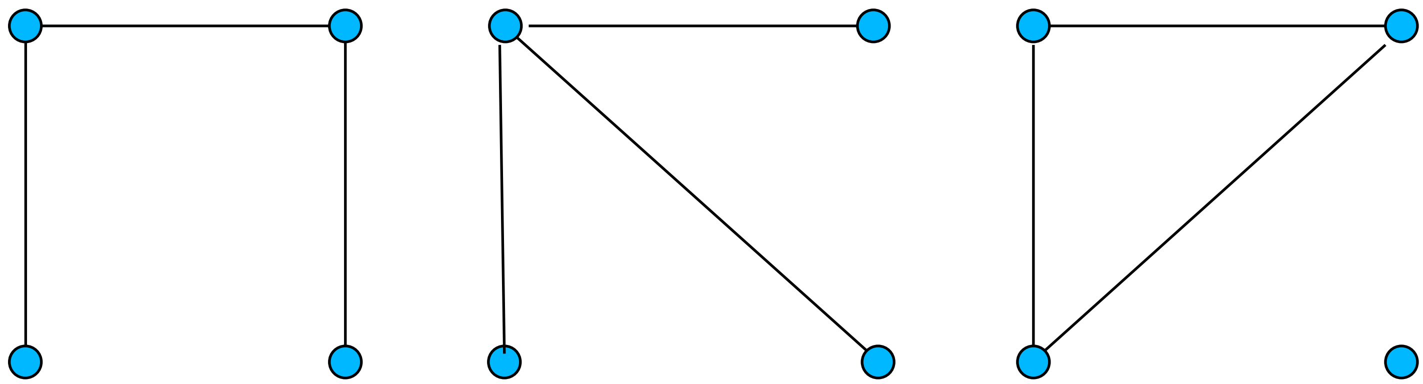
Basic properties of graphs

- This is a necessary, but not sufficient condition for two graphs to be isomorphic
- However, if at least one vertex of G_1 is not equal to the degree of vertex G_2 then they are definitely not isomorphic
- Single graph determines equivalence class
  - There is only one graph of order 1 (size zero); there is only one graph of order 2 and size 0 or size 1
  - There are 4 graphs of order 4 and size 3



Basic properties of graphs

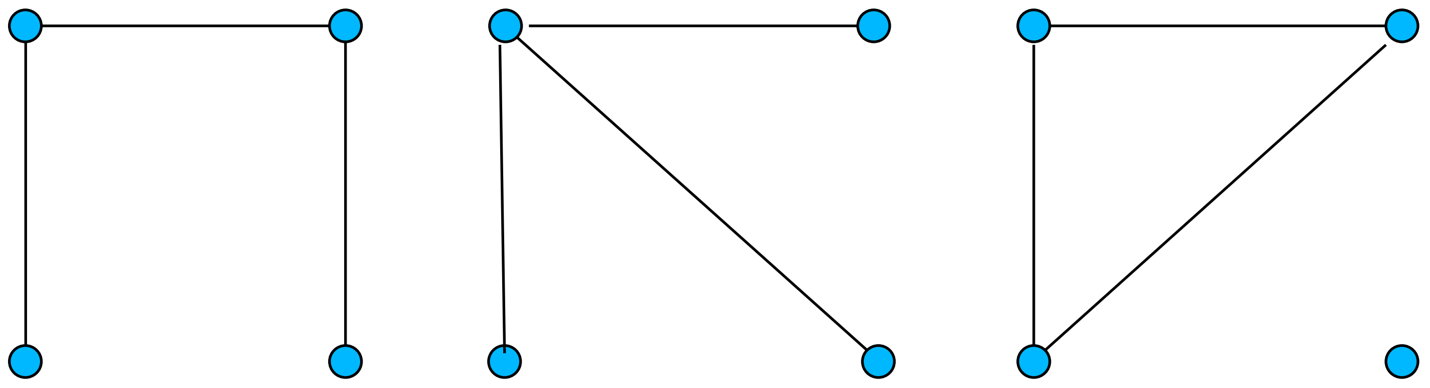
- This is a necessary, but not sufficient condition for two graphs to be isomorphic
- However, if at least one vertex of G_1 is not equal to the degree of vertex G_2 then they are definitely not isomorphic
- Single graph determines equivalence class
  - There is only one graph of order 1 (size zero); there is only one graph of order 2 and size 0 or size 1
  - There are 4 graphs of order 4 and size 3



Basic properties of graphs

- If we have 4 graphs of size 4 and order 3, at least 2 of them will be in the same equivalence class
- **The Pigeonhole principle**:

  Let set S be s.t. $|S|=n$ and $S_1,\ldots,S_k$ are partitions of S. Then at least one set S_i contains at least $\{n / k\}$ elements



Basic properties of graphs

- Graph H is a subgraph of G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$
- For $u, v \in V(G)$, u-v walk in G is a sequences of vertices and edges that starts in u and ends in v
- For $u, v \in V(G)$, u-v trail in G is a u-v walk which does not repeat any edge; u-v path in G is a u-v walk which does not repeat any vertex
- Vertices $u, v \in V(G)$ are connected if either $u=v$ or $u \neq v$ and there exists u-v path in G

Basic properties of graphs

- Graph G is connected if every two vertices of G are connected (otherwise it is disconnected)
- A connected subgraph of G is called a component of G if it is not contained in any connected subgraph of G
- u-u trail with at least 3 edges is called a circuit
- A circuit that does not repeat any vertices is called a cycle
- Let $G \setminus e$ ($G \setminus v$) be the graph constructed by removing edge e (vertex v) from graph G
  - $e \in E(G)$ is a bridge if $G \setminus e$ is disconnected
  - $v \in V(G)$ is a cut vertex if $G \setminus v$ is disconnected

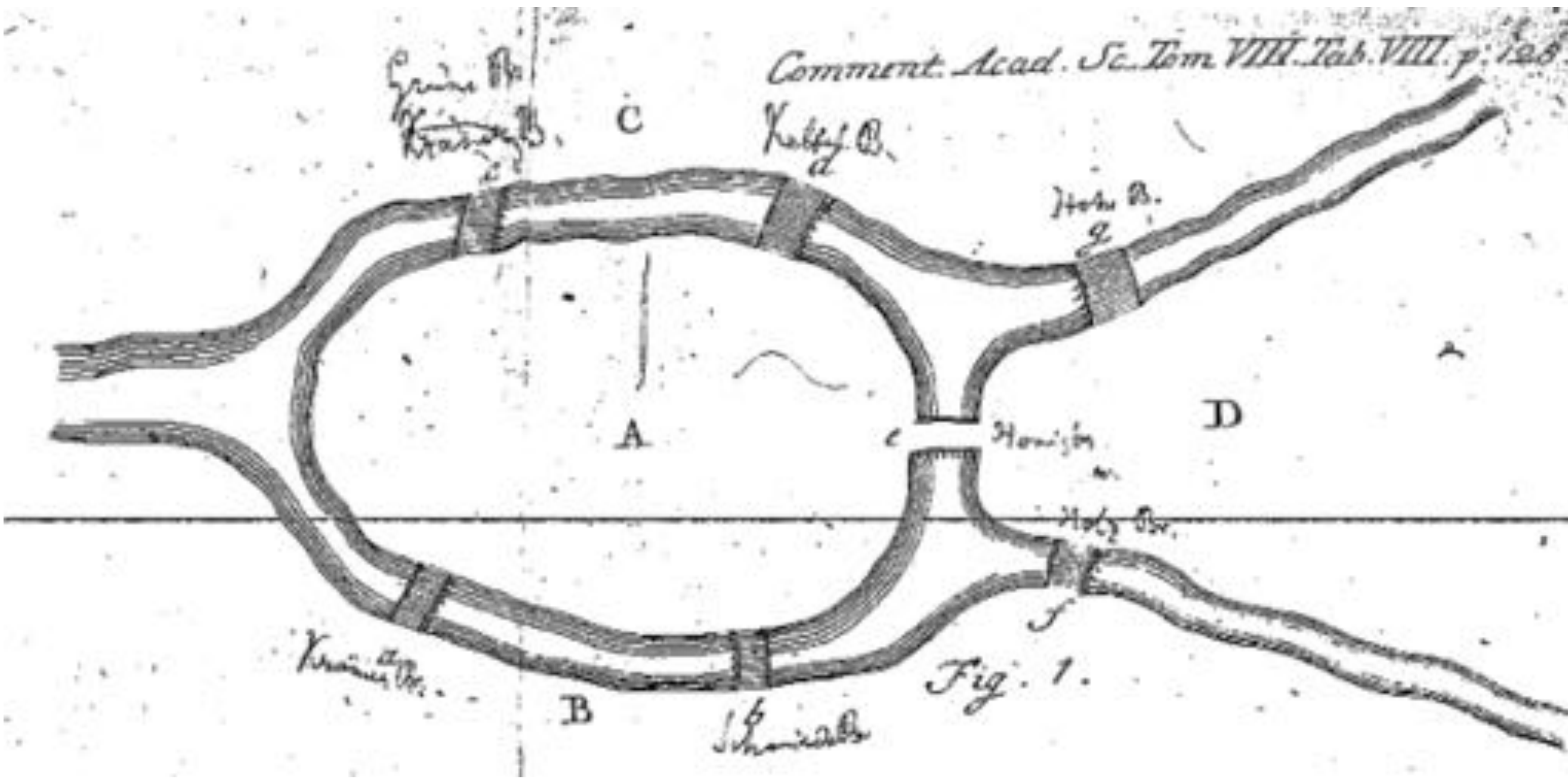
Basic properties of graphs

Theorem

Let G be a connected graph. An edge e of G is a bridge of G iff e does not lie on any cycle of G

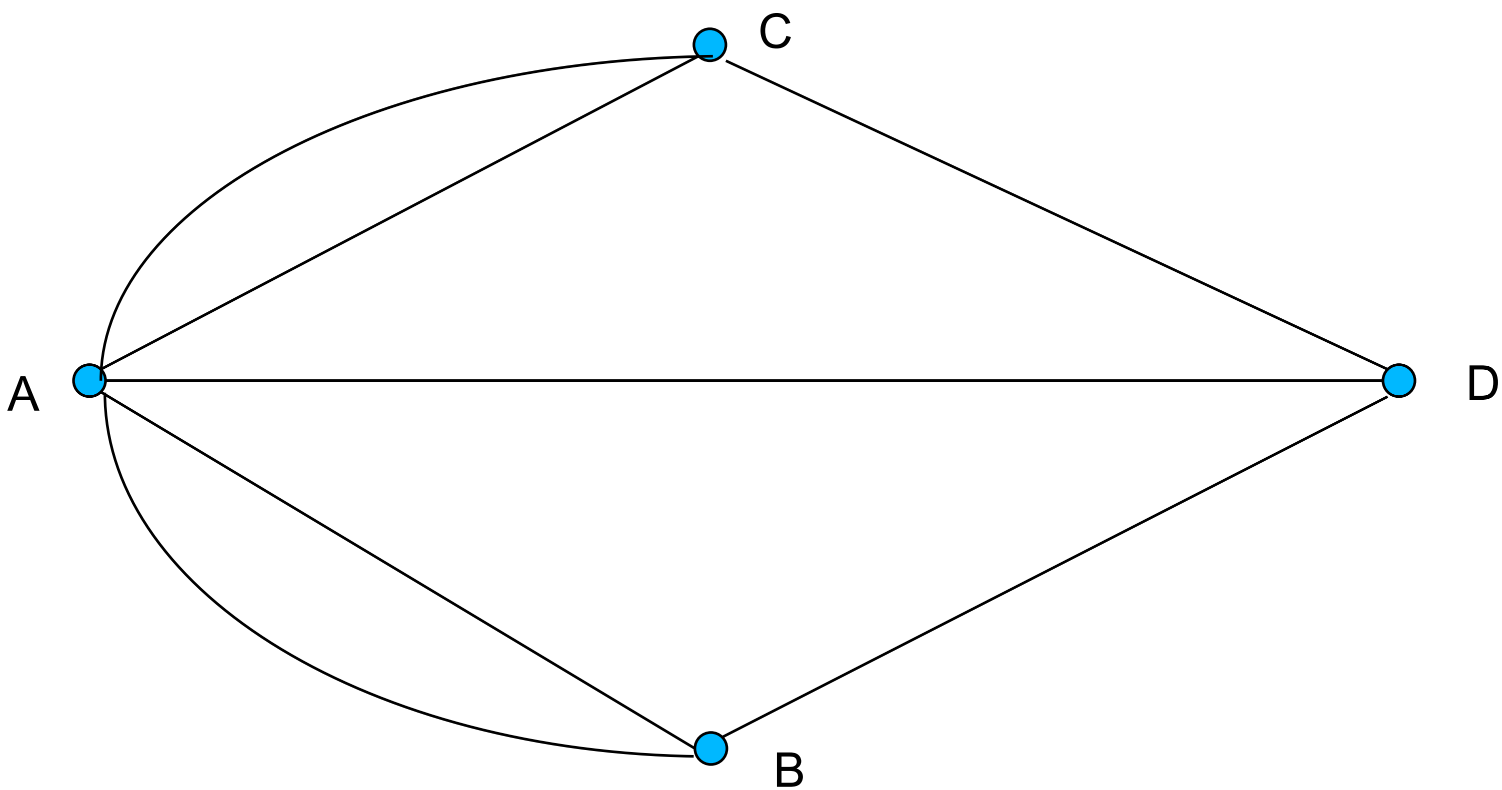
The Königsberg Bridge problem

- Is it possible to cross all seven bridges in a continuous walk without repeating any of them?



The Königsberg Bridge problem

- Is it possible to cross all seven bridges in a continuous walk without repeating any of them?



The Königsberg Bridge problem

- Circuit containing all vertices and edges of graph *G* is called eulerian circuit in *G*
- Graph *G* containing an eulerian circuit is called eulerian graph

Theorem

Graph G is eulerian iff G is connected and every vertex of G is even

Corollary

The Königsberg Bridge problem has no solutions

The salesman problem

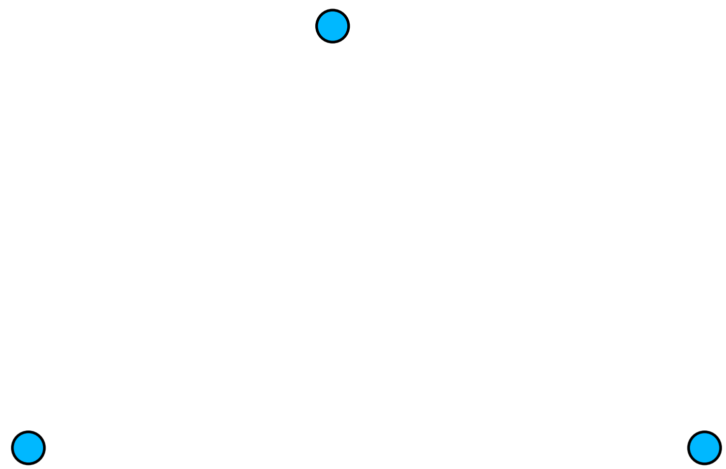
- “Transportation network” links cities by highways. Can a salesman create a connected path that visits each city once?
- Graph G is hamiltonian if a cycle that contains every vertex exists in G. The cycle containing all vertices of G is referred to as hamiltonian cycle.

Theorem

If G is of order p (≥ 3) such that $\deg(v) \geq p/2$ for all $v \in V(G)$, then G is Hamiltonian.

The salesman problem

- **Illustration**: If p=3 and deg(v) ≥3/2 and, thus deg(v)=2

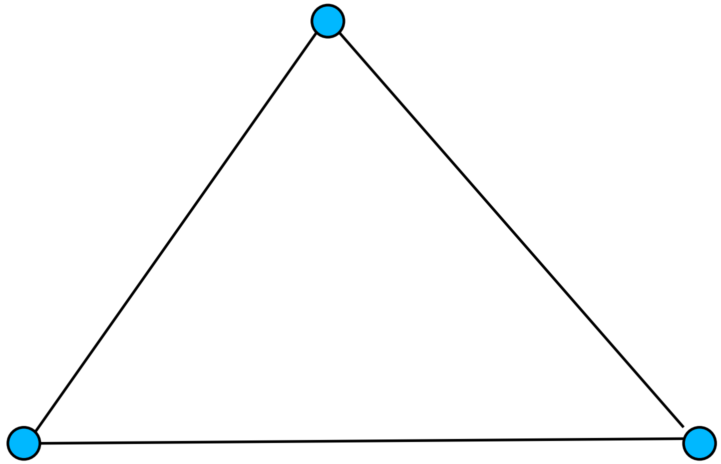


Theorem

If G is of order p (≥3) such that deg(v) ≥ p/2 for all v ∈ V(G) , then G is Hamiltonian.

The salesman problem

- **Illustration**: If p=3 and deg(v) ≥3/2 and, thus deg(v)=2

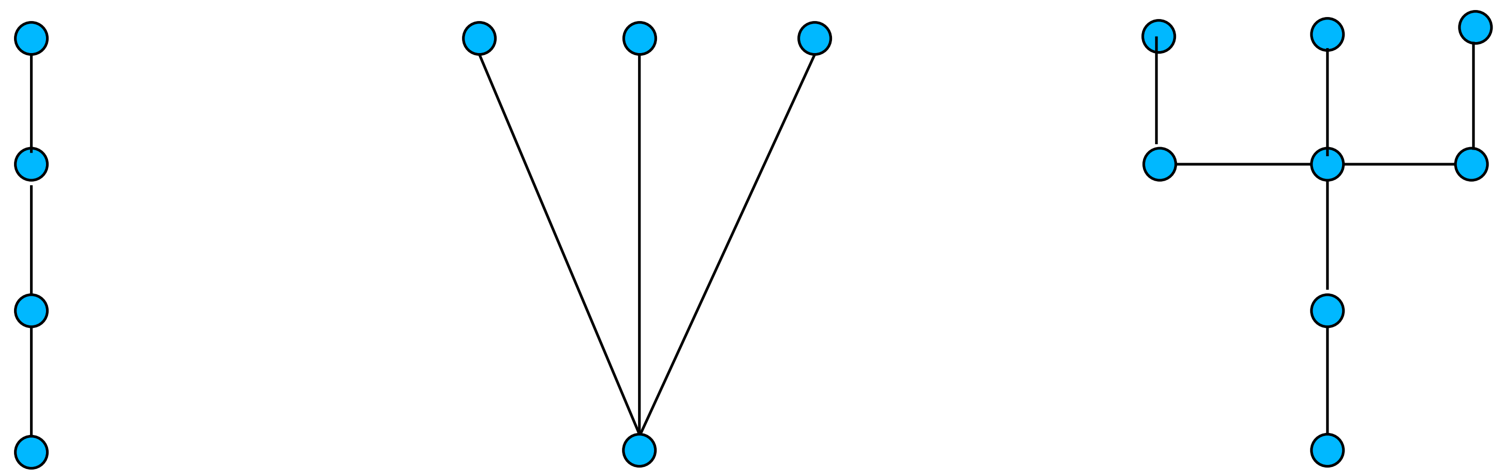


Theorem

If G is of order p (≥3) such that deg(v) ≥ p/2 for all $v \in V(G)$, then G is Hamiltonian.

Trees

- Any connected graph with no cycles is called a tree
- Every edge of the tree is a bridge
- A graph without cycles (but not necessarily connected) is called a forest



Trees

- Any connected graph with no cycles is called a tree
- Every edge of the tree is a bridge
- A graph without cycles (but not necessarily connected) is called a forest

Theorem

Suppose that G is a tree and $v, u \in V(G)$, then there exists a unique u-v path in G.

Theorem

Suppose that G is a tree of order p and size q, then $q = p - 1$.

Trees

- Consider undirected network with graph *G*. Spanning tree *T* of *G* is a tree that is a subgraph of *G* that contains all vertices of *G*.
- Greedy tree constructed from G by sequentially selecting edges of network with smallest value that do not form cycle with previously selected edges

Theorem

Consider undirected network with graph *G* and let *T* be a greedy tree of *G.* Then *T* is the spanning tree of *G* with the smallest value.

Trees

- Greedy trees are very important in Machine Learning
- In linear regression, a real-valued dependent variable Y is modeled as a linear function of a real-valued independent variable X plus noise:

  $Y = a_0 + a_1 X + u$

- In multiple regression there are multiple independent variables

  $Y = a_0 + a_1 X_1 + \ldots + a_k X_k + u$

- This model is "tractable" as long as it is separable
- May be necessary to incorporate interactions to capture nonlinearities
- The number of parameters is clearly getting very large very fast with even two-way interactions among the independent variables, and stronger nonlinearities are going to be trouble.

Trees

- Linear regression is a global model, where there is a single predictive formula holding over the entire data-space.
- When the data has lots of features which interact in complicated, nonlinear ways, assembling a single global model can be very difficult, and hopelessly confusing when you do succeed.
- An alternative approach to nonlinear regression is to sub-divide, or partition, the space into smaller regions, where the interactions are more manageable.
- We then partition the sub-divisions again this is called recursive partitioning until finally we get to chunks of the space which are so small that we can fit simple models to them.
- The global model thus has two parts: one is just the recursive partition, the other is a simple model for each cell of the partition.

Trees

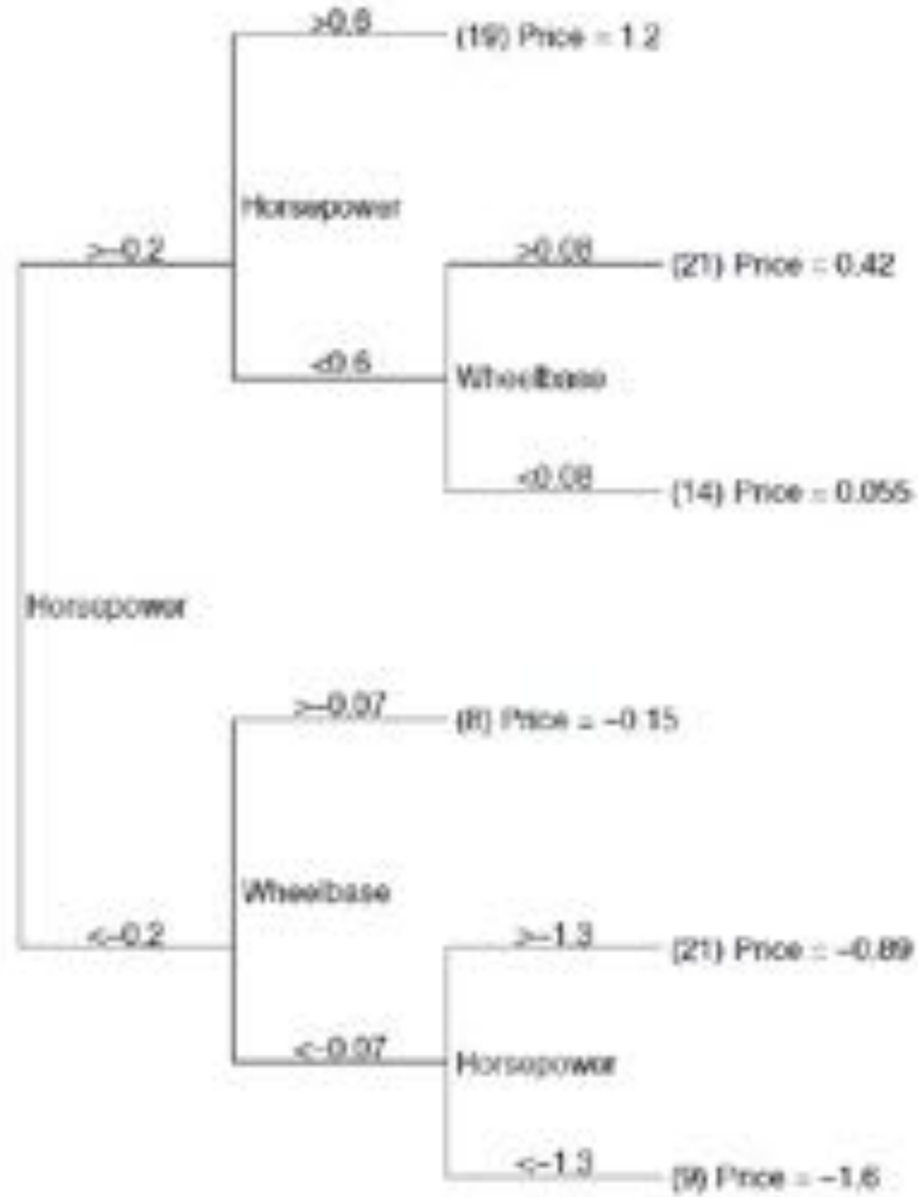
- Algorithm originates from Breiman (2001)
- Task: nonparametric estimation with feature vector $X \in \boldsymbol{R}^p$
- Goal: predict outcome $Y \in \boldsymbol{R}$
- By estimating regression function $m(x) = \mathbf{E}[\, Y \mid X = x \,]$
- Approach:
  - Define graph by placing vertices at centers of rectangular partition of space
  - Use (very) simple prediction: average of Y within each rectangle (leaf)
  - Define cost as the mean-squared error
  - Construct a greedy tree

Trees

- To figure out which cell we are in, we start at the root node of the tree, and ask a sequence of questions about the features.
- The interior nodes are labeled with questions, and the edges or branches between them labeled by the answers.
- Which question we ask next depends on the answers to previous questions.
- In the classic version, each question refers to only a single attribute, and has a yes or no answer
  - Is Horsepower > 50?
  - Is EconStudent == FALSE?
- Variables do not all have to be of the same type; some can be continuous, some can be discrete but ordered, some can be categorical, etc.
- You could do more-than-binary questions, but that can always be accommodated as a larger binary tree.

Trees

Predicting prices of cars in 1993

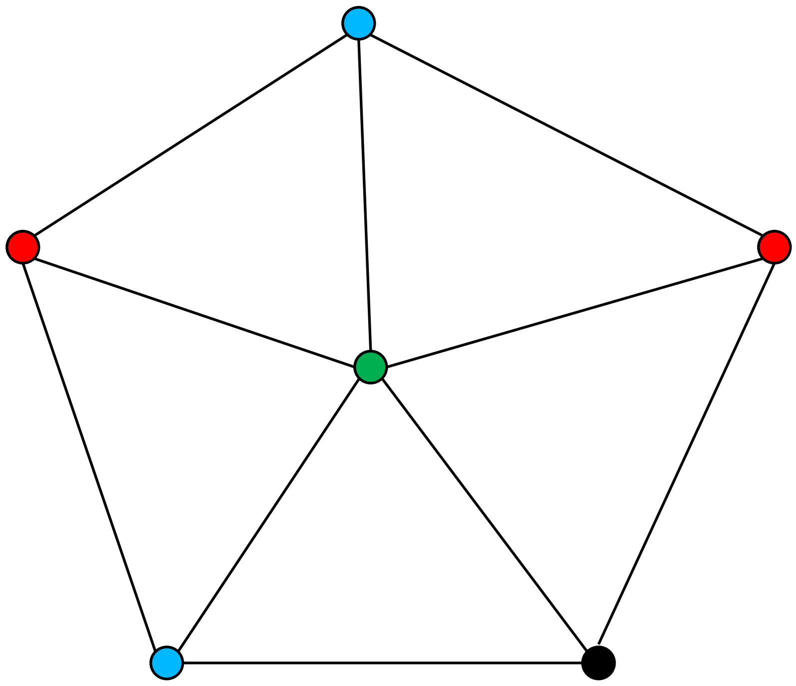


Coloring problems

- Many important computational problems reduce to colorings on graphs
- Coloring of graph G is assignment of colors to vertices of G so that adjacent vertices have different color (n-coloring of G uses n colors)
- It is always possible to produce p-coloring of graph of order p
- Chromatic number of graph G (denoted $\chi(G)$) is minimum value n for which n-coloring of G exists

Coloring problems

- Example: Prove that $\chi(G)=4$



Coloring problems

- **Scheduling problem**: create a schedule of courses so that no two classes meet at the same time (if student wants to take them) with minimum length
- Graph G has vertices corresponding to classes; two vertices are connected if there is a student who wants to take both classes

Theorem

The minimum length of the schedule in the scheduling problem is $\chi(G)$

Coloring problems

- Finding chromatic numbers is difficult
- In fact, one approach to characterize difficulty of computational problems is by reduction to coloring problems
- However, approximation may be easier

Theorem

For any graph G

$$\chi(G) \leq 1 + \max_{v \in V(G)} \deg(v)$$