Week 12 – Wednesday

COMP 2230

Last time

- Conditional probability

Questions?

Assignment 6

Logical warmup

- A man offers you a bet
- He shows you three cards
  - One is red on both sides
  - One is green on both sides
  - One is red on one side and green on the other



- He will put one of the cards, at random, on the table, with a random side up
- If you can guess the color on the other side, you win
- If you bet $100
  - You gain $60 on a win
  - You lose your $100 on a loss
- Should you take the bet? Why or why not?

Review for Exam 3

Graphs and Trees

Graphs

- A **graph** G is made up of two finite sets
  - **Vertices**: $V(G)$
  - **Edges**: $E(G)$
- Each edge is connected to either one or two vertices called its **endpoints**
- An edge with a single endpoint is called a **loop**
- Two edges with the same sets of endpoints are called **parallel**
- Edges are said to **connect** their endpoints
- Two vertices that share an edge are said to be **adjacent**
- A graph with no edges is called **empty**

Special graphs

- A **simple graph** does not have any loops or parallel edges
- Let n be a positive integer
- A **complete graph** on n vertices, written K_n, is a simple graph with n vertices such that every pair of vertices is connected by an edge
- A **bipartite graph** is a simple graph whose vertices can be partitioned into sets X and Y such that there are no edges from any vertex in X to another vertex in X and there are no edges from any vertex in Y to another vertex in Y
- A **complete bipartite graph on** (m, n) **vertices**, written $K_{m,n}$ is a simple graph with a set of m vertices and a disjoint set of n vertices such that:
  - There is an edge from each of the m vertices to each of the n vertices
  - There are no edges among the set of m vertices
  - There are no edges among the set of n vertices
- A **subgraph** is a graph whose vertices and edges are a subset of another graph

Degree

- The **degree of a vertex** is the number of edges that are incident on the vertex
- The **total degree of a graph** G is the sum of the degrees of all of its vertices
- What's the relationship between the degree of a simple graph and the number of edges it has?
- What's the degree of a complete graph with n vertices?
- Note that the number of vertices with odd degree must be even

Example

- What is the maximum number of edges a simple disconnected graph with n vertices can have?
- Prove it

Definitions

- A **walk from** v **to** w is a finite alternating sequence of adjacent vertices and edges of G, starting at vertex v and ending at vertex w
  - A walk must begin and end at a vertex
- A **path from** v **to** w is a walk that does not contain a repeated edge
- A **simple path from** v **to** w is a path that does not contain a repeated vertex
- A **closed walk** is a walk that starts and ends at the same vertex
- A **circuit** is a closed walk that does not contain a repeated edge
- A **simple circuit** is a circuit that does not have a repeated vertex other than the first and last

Circuits

- If a graph is connected, non-empty, and every node in the graph has even degree, the graph has an **Euler circuit**
- Algorithm to find one:
  1. Pick an arbitrary starting vertex
  2. Move to an adjacent vertex and remove the edge you cross from the graph
     - Whenever you choose such a vertex, pick an edge that will not disconnected the graph
  3. If there are still uncrossed edges, go back to Step 2
- A **Hamiltonian circuit** must visit every vertex of a graph exactly once (except for the first and the last)
- If a graph G has a Hamiltonian circuit, then G has a subgraph H with the following properties:
  - H contains every vertex of G
  - H is connected
  - H has the same number of edges as vertices
  - Every vertex of H has degree 2

Matrix representations of graphs

- To represent a graph as an **adjacency matrix**, make an $n \times n$ matrix a, where n is the number of vertices
- Let the nonnegative integer at a_{ij} give the number of edges from vertex i to vertex j
- A simple graph will always have either 1 or 0 for every location
- We can find the number of walks of length k that connect two vertices in a graph by raising the adjacency matrix of the graph to the k^{th} power

Isomorphism

- We call two graphs isomorphic if we can relabel one to be the other
- Graph isomorphisms define equivalence classes
  - **Reflexive:** Any graph is clearly isomorphic to itself
  - **Symmetric:** If x is isomorphic to y, then y must be isomorphic to x
  - **Transitive:** If x is isomorphic to y and y is isomorphic to z, it must be the case that x is isomorphic to z

Isomorphism invariants

- A property is called an **isomorphism invariant** if its truth or falsehood does not change when considering a different (but isomorphic) graph
- These **cannot** be used to prove isomorphism, but they can be used to disprove isomorphism
- 10 common isomorphism invariants:
  1. Has n vertices
  2. Has m edges
  3. Has a vertex of degree k
  4. Has m vertices of degree k
  5. Has a circuit of length k
  6. Has a simple circuit of length k
  7. Has m simple circuits of length k
  8. Is connected
  9. Has an Euler circuit
  10. Has a Hamiltonian circuit

Isomorphism practice

- Draw two graphs with 7 nodes and 10 edges in which the list of degrees on the first graph is identical to the list of degrees on the second, yet the graphs are still not isomorphic

Trees

- A **tree** is a graph that is **circuit-free** and **connected**
- Any tree with more than one vertex has at least two vertices of degree 1
- If a vertex in a tree has degree 1 it is called a **terminal vertex** (or **leaf**)
- All vertices of degree greater than 1 in a tree are called **internal vertices** (or **branch vertices**)
- For any positive integer n, a tree with n vertices must have $n - 1$ edges

Rooted trees

- In a **rooted tree**, one vertex is distinguished and called the **root**
- The **level** of a vertex is the number of edges along the unique path between it and the root
- The **height** of a rooted tree is the maximum level of any vertex of the tree
- The **children** of any vertex v in a rooted tree are all those nodes that are adjacent to v and one level further away from the root than v
- Two distinct vertices that are children of the same parent are called **siblings**
- If w is a child of v, then v is the parent of w
- If v is on the unique path from w to the root, then v is an ancestor of w and w is a descendant of v

Binary trees

- A **binary tree** is a rooted tree in which every parent has at most two children
- Each child is designated either the **left child** or the **right child**
- In a **full binary tree**, each parent has exactly two children
- Given a parent v in a binary tree, the **left subtree** of v is the binary tree whose root is the left child of v
- Ditto for **right subtree**

Spanning Trees

Spanning trees

- A **spanning tree** for a graph G is a subgraph of G that contains every vertex of G and is a tree
- Some properties:
  - Every connected graph has a spanning tree
    - Why?
  - Any two spanning trees for a graph have the same number of edges
    - Why?

Weighted graphs

- In computer science, we often talk about **weighted graphs** when tackling practical applications
- A weighted graph is a graph for which each edge has a real number **weight**
- The sum of the weights of all the edges is the **total weight** of the graph
- Notation: If e is an edge in graph G, then $w(e)$ is the weight of e and $w(G)$ is the total weight of G
- A **minimum spanning tree** (**MST**) is a spanning tree of lowest possible total weight

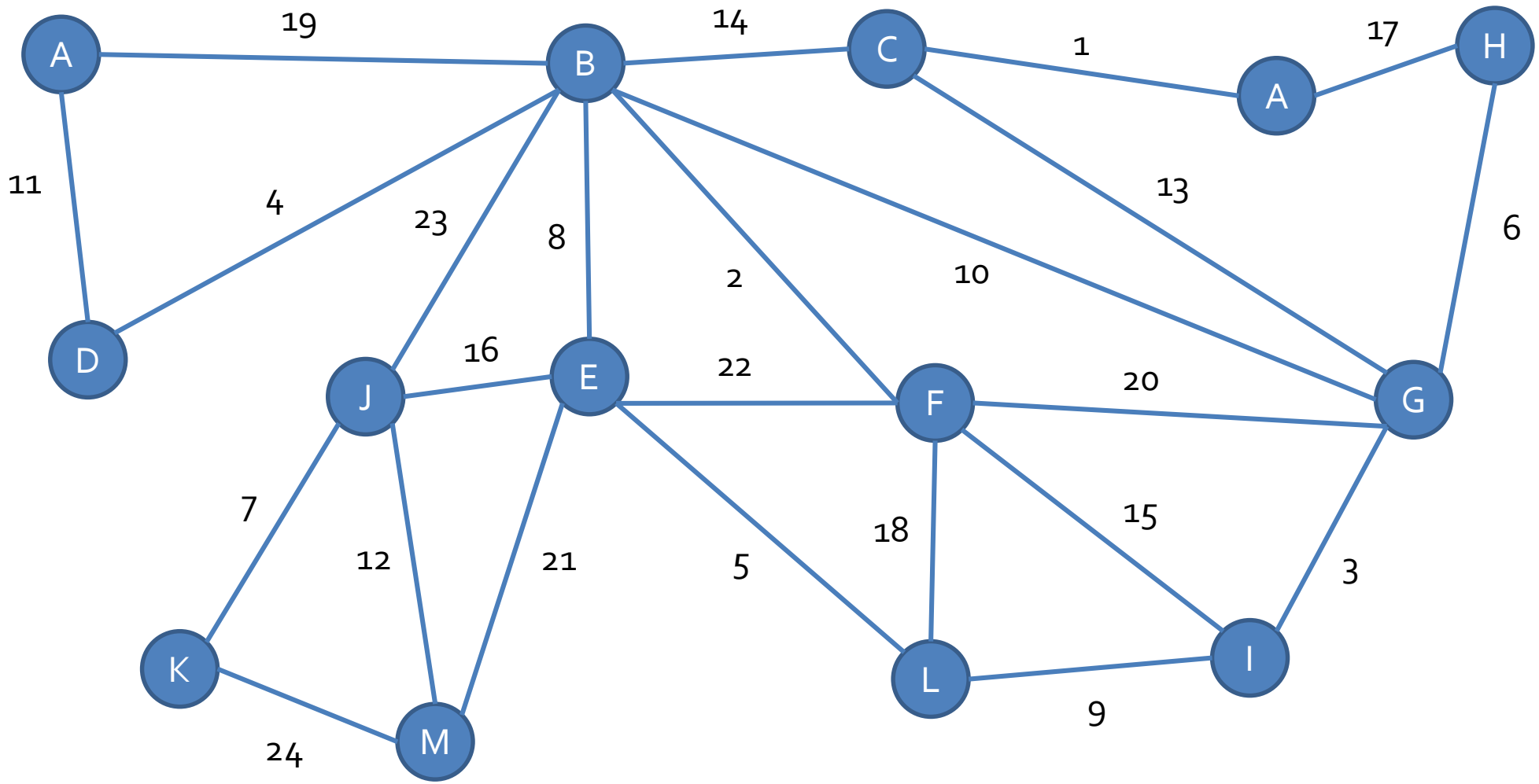
Finding a minimum spanning tree

- Kruskal's algorithm gives an easy to follow technique for finding an MST on a weighted, connected graph
- Informally, go through the edges, adding the smallest one, unless it forms a circuit
- **Algorithm:**
  - Input: Graph G with n vertices
  - Create a subgraph T with all the vertices of G (but no edges)
  - Let E be the set of all edges in G
  - Set $m = 0$
  - While $m < n - 1$
    - Find an edge e in E of least weight
    - Delete e from E
    - If adding e to T doesn't make a circuit
      - Add e to T
      - Set $m = m + 1$
  - Output: T

Prim's algorithm

- Prim's algorithm gives another way to find an MST
- Informally, start at a vertex and add the next closest node not already in the MST
- **Algorithm:**
  - Input: Graph G with n vertices
  - Let subgraph T contain a single vertex v from G
  - Let V be the set of all vertices in G except for v
  - For i from 1 to $n - 1$
    - Find an edge e in G such that:
      - e connects T to one of the vertices in V
      - e has the lowest weight of all such edges
    - Let w be the endpoint of e in V
    - Add e and w to T
    - Delete w from V
  - Output: T

MST practice



Probability and Counting

Multiplication rule

- If an operation has k steps such that
  - Step 1 can be performed in n_1 ways
  - Step 2 can be performed in n_2 ways

    …
  - Step k can be performed in n_k ways
- Then, the entire operation can be performed in $\prod_{i=1}^{k} n_i = n_1 n_2 \ldots n_k$ ways
- This rule only applies when each step always takes the same number of ways (unlike the previous possibility tree example)

Addition and inclusion/exclusion rules

- If a finite set A equals the union of k distinct mutually disjoint subsets A_1, A_2, … A_k, then:

$$N(A) = N(A_1) + N(A_2) + \ldots + N(A_k)$$

- If A, B, C are any finite sets, then

$$N(A \cup B) = N(A) + N(B) - N(A \cap B)$$

- And

$$N(A \cup B \cup C) = N(A) + N(B) + N(C) - N(A \cap B) - N(A \cap C) - N(B \cap C) + N(A \cap B \cap C)$$

Pigeonhole principle

- If n pigeons fly into m pigeonholes, where $n > m$, then there is at least one pigeonhole with two or more pigeons in it
- More formally, if a function has a larger domain than co-domain, it cannot be one-to-one
- We cannot say exactly how many pigeons are in any given holes
- Some holes may be empty
- But at least one hole will have at least two pigeons

Generalized pigeonhole principle

- If n pigeons fly into m pigeonholes, and for some positive integer k, $n > km$, then at least one pigeonhole contains $k + 1$ or more pigeons in it
- Example:
  - In a group of 85 people, at least 4 must have the same last initial

Handy dandy guide to counting

- This is a quick reminder of all the different ways you can count things:

	Order Matters	Order Doesn't Matter
Repetition Allowed	n^k	$\binom{k+n-1}{k}$
Repetition Not Allowed	$P(n,k)$	$\binom{n}{k}$

How many ways are there to…

- Order the letters in the word "replicant"
- Order any three letters in the word "replicant"?
- Order the letters in the word "neverness"
  - If all letters are considered distinct?
  - If identical letters cannot be told apart?
- Select an elite assassin team of 8 people from a squadron of 100?
- Select an elite assassin team of 8 people from a squadron of 100 and make one person the leader?

How many ways are there to…

- Name a child if you're willing use any sequence of characters from the Latin alphabet from 1 to 10 characters long?
- Decide the five courses you'd like to take in a semester, if the kinds are:
  - Computer Science courses
  - INST courses
  - Electives

Binomial theorem

- $a + b$ is called a **binomial**
- Using combinations (or Pascal's Triangle) gives an easy way to compute $(a + b)^n$

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^{n-k} b^k$$

Binomial practice

- Thoroughly simplify $\left(a - \frac{1}{2}b\right)^6$

Probability axioms

- Let A and B be events in the sample space S
  - $0 \leq P(A) \leq 1$
  - $P(\emptyset) = 0$ and $P(S) = 1$
  - If $A \cap B = \emptyset$, then $P(A \cup B) = P(A) + P(B)$
  - It is clear then that $P(A^c) = 1 - P(A)$
  - More generally, $P(A \cup B) = P(A) + P(B) - P(A \cap B)$
- All these axioms can be derived from set theory and the definition of probability

Expected value

- **Expected value** is one of the most important concepts in probability, especially if you want to gamble
- The expected value is simply the sum of all events, weighted by their probabilities
- If you have n outcomes with real number values $a_1, a_2, a_3, \ldots\ a_n$, each of which has probability $p_1, p_2, p_3, \ldots\ p_n$, then the expected value is:

$$\sum_{k=1}^{n} a_k p_k$$

Conditional probability

- Given that some event A has happened, the probability that some event B will happen is called conditional probability
- This probability is:

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

Upcoming

Next time…

- Exam 3

Reminders

- Keep working on Assignment 6
  - Due Friday
- Study for Exam 3
  - Also on Friday