CIS 1210—Data Structures and Algorithms—Spring 2025

Review—Tuesday, January 21 / Wednesday, January 22

Learning Goals

During this recitation, you will:

- Review common proof techniques
- Build intuition for the connection between induction and recursion
- Review some graph theory

Proof Techniques

Fundamental proof techniques. For contradiction and contrapositive, try writing out the truth tables to prove to yourself that these statements are logically equivalent to a direct proof.

- Direct: $p \implies q$
- Contrapositive: $\neg q \implies \neg p$
- Contradiction: $p \wedge \neg q \implies C$
- Induction: To prove $P(n)\ \forall n \geq b$, show $P(b)$ and $\forall n \geq b$, $P(n) \implies P(n+1)$

Graphs

A graph $G = (V, E)$ consists of a set of vertices (also called nodes) V together with a set of edges $E \subseteq V \times V$, where $V \times V$ denotes the Cartesian product of V with itself. We denote an edge from u to v as (u, v). A graph can be:

- **Undirected**: The edges (u, v) and (v, u) are considered the same.
- **Directed**: The edges (u, v) and (v, u) are distinct.

In CIS 1210, we do not consider self-loops (i.e., edges of the form (v, v)). The **degree** of a vertex v, denoted $\deg(v)$, is the number of edges incident on it. For directed graphs, we distinguish between:

- **In-degree**: Number of edges directed into the vertex.
- **Out-degree**: Number of edges directed out from the vertex.

Paths and Cycles

A **path** from vertex u to v is a sequence:

$$u = v_0, v_1, \ldots, v_n = v$$

such that $(v_i, v_{i+1}) \in E$ for all $0 \leq i < n$. The length of a path is the number of edges it contains. A **simple path** contains distinct vertices, meaning $v_i \neq v_j$ for all $i \neq j$.

A **cycle** is a sequence:

$$v_0, v_1, \ldots, v_n = v_0$$

where the first and last vertices are the same. Graphs that contain cycles are called **cyclic**, while those that do not are called **acyclic**.

Connectedness

A graph is said to be:

- **Connected** (for undirected graphs): There is a path between any two vertices.
- **Strongly connected** (for directed graphs): There is a path from u to v and from v to u.

A (strongly) connected component of a graph is a maximal subgraph that is (strongly) connected.

Special Types of Graphs

A **tree** is an undirected, connected, acyclic graph. Some defining properties of trees include:

- A tree with n nodes is connected and has exactly $n - 1$ edges.
- There is a unique path between any pair of vertices.
- Trees are **minimally connected**: removing any edge disconnects the tree.
- Adding any edge to a tree creates a cycle.

A vertex with degree 1 is called a **leaf**. A **forest** is a disjoint collection of trees.

Binary Trees

A **binary tree** is a tree in which every vertex has at most two children. Typically, the tree is rooted at a specific node, and the terms left and right children are used to describe neighbors. Important properties include:

- The height of a binary tree is the number of edges in the longest path from the root to a leaf.
- The ancestors, descendants, and parents of nodes are defined as expected.

Random Variables

Definition. A random variable X on a sample space Ω is a real-valued function that assigns to each sample point $\omega \in \Omega$ a real number $X(\omega)$.

Intuitively you can think of a random variable as mapping an event in your sample space to some value. For example, if X represents the number of heads after flipping 2 coins, X maps the event (H, T) to the number 1.

Note, in CIS 1210, you do not need to explicitly define the sample space when dealing with random variables.

Definition. Expectation of a random variable denoted $\mathbb{E}[X]$ is

$$\mathbb{E}[X] = \sum_i i \cdot \Pr[X = i]$$

We can think of the expectation to be what value we "expect" X to take on after many repeated trials.

Linearity of Expectation. For any finite collection of random variables $X_1, X_2, \ldots, X_n$

$$\mathbb{E}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathbb{E}[X_i]$$