*Graphs:*

- Representing a relation by graphs. \( V, R \subseteq V \times V \), say \( V \) is the set of vertices and \( R \) is the set of edges.

  - binary edge
  - multiedge

*Ex:*

- Family represented in a graph.
  - This is a family with 4 people.

*Ex:*

- \( (\text{name, email, web}) \) ≠ not a binary relation
  
  - 1: (Luca, Luca, MLuca)
  - 2: (John, Luca, MLJ)

- Can always convert into binary structure(edges):

  - \( A \rightarrow B \Rightarrow \square \rightarrow \square \rightarrow \square \)

*Ex:* Wiring a circuit board:

  - Problem: Minimize number of planes so wires are not crossed (among many other restraints)

  - Problem: Minimum Cut: removing minimum number of edges so the graph is broken into two or more disconnected subgraphs.

  - Problem: Dynamic Graphs: Shortest path on a dynamic path.

In Luca's work, graphs are very large. The algorithms we learn in this class are generally too slow in practice (exponential or worse run time). Heuristics are used to create feasible algorithms for real programs.
Definitions

- Graph $G = (V, E, \rho)$
  - $e \in E$, $(u, v) \in V \times V = V^2$
  - $\rho: E \rightarrow V \times V$, $\rho(eeE) \rightarrow (u, v) \in V^2$
  - $\rho_s(e)$ = source $\in V$
  - $\rho_d(e)$ = destination $\in V$
  - $\rho(e) = (\rho_s(e), \rho_d(e))$

Directed undirected edges

Directed

- MultiGraph

- Undirected

- Clark, Holton (Theory Book)

Graph

- Cormen (Algorithm book)

Definitions:

- **Parallel edges**: $e, f \in E$ are parallel iff $\rho(e) = \rho(f)$
- **Self loop**: $e \in E$ is a self loop iff $\rho_s(e) = \rho_d(e)$
- **Simple Graph**: No parallel edges and no self loops.

Isomorphism: $G_1 = (V_1, E_1, \rho_1)$ and $G_2 = (V_2, E_2, \rho_2)$. $G_1$ and $G_2$ are isomorphic iff:

- there is: $\forall e \in E_1 \times E_2$

Note: Bijection implies $|V_1| = |V_2|$ and $|E_1| = |E_2|$.

- if $e_1 \in E_2$, then $\rho_s(e_1) \sim \rho(s(e_2))$ and $\rho_d(e_1) \sim \rho(d(e_2))$

- if $v_1 \sim v_2$ then... (implied by edges)
Problem: Algorithms solving whether graphs are isomorphic are good in average case but too slow in worst case.

Definitions:

Labels: \( l: V \rightarrow \text{label} \quad l(v) = \text{label}_v \)

Bi-Similar: \( G_1 = (V_1, E_1, \alpha_1) \) and \( G_2 = (V_2, E_2, \alpha_2) \)
\( \sim \in V_1 \times V_2 \) ST if \( V_1 \sim V_2 \) then:
1. \( l(v_1) = l(v_2) \)
2. for \( \forall e \) with \( \alpha_1(e) = v_1 \), there is \( f \) with \( \alpha_2(f) = v_2 \)
\( ST \ l(e) = l(f) \) and \( \beta_1(e) = \beta_2(f) \)
3. for \( \forall f \) with \( \alpha_2(f) = v_2 \) ... symmetrical with (2)

Problem: Find largest bi-simulation.

Ex:

\[ G_1 \xrightarrow{R} G \]
\[ G \xrightarrow{R} G_2 \]
\[ G_2 \xrightarrow{R} \]
\[ G \xrightarrow{R} G_3 \]

\( G_1 \) and \( G_2 \) not bi-similar.
\( G_1 \) and \( G_3 \) are bi-similar. (the largest bi-similarity)

Note: Bi-similarity does not refer necessarily to entire graphs; \( G_j \) bi-similar to \( G_k \) may mean a subgraph of \( G_k \) is bi-similar to \( G_j \).