VII. Graph Algorithms
Notion of graphs

Basic terminology

- Graph $G = (V, E)$:
  - $V = \{v_i\}$ = set of vertices
  - $E$ = set of edges = a subset of $V \times V = \{(v_i, v_j)\}$

- $|E| = O(|V|^2)$
  - dense graph: $|E| \approx |V|^2$
  - sparse graph: $|E| \approx |V|$

- Some variants
  - undirected: edge $(u, v) = (v, u)$
  - directed: $(u, v)$ is edge from $u$ to $v$.
  - weighted: weight on either edge or vertex
  - multigraph: multiple edges between vertices

- Reading: Appendix B.4, pp.1168-1172 of [CLRS,3rd ed.]
Notion of graphs

Representing a graph by an **Adjacency Matrix**

- \( A = (a_{ij}) \) is a \(|V| \times |V|\) matrix, where
  \[
  a_{ij} = \begin{cases} 
  1, & \text{if } (v_i, v_j) \in E \\
  0, & \text{otherwise}
  \end{cases}
  \]

- If \( G \) is undirected, \( A \) is symmetric, i.e., \( A^T = A \).
- \( A \) is typically very sparse – use a sparse storage scheme in practice.
Notion of graphs

Representing a graph by an **Incidence Matrix**

- $B = (b_{ij})$ is a $|V| \times |E|$ matrix, where

$$b_{ij} = \begin{cases} 
1, & \text{if edge } e_j \text{ enters vertex } v_i \\
-1, & \text{if edge } e_j \text{ leaves vertex } v_i \\
0, & \text{otherwise}
\end{cases}$$
Notion of graphs

Representing a graph by an Adjacency List

- For each vertex \( v \),
  \[
  \text{Adj}[v] = \{ \text{vertices adjacent to } v \}
  \]
- Variation: could also keep second list of edges coming into vertex.
- How much storage is needed?
  Answer: \( \Theta(|V| + |E|) \) ("sparse representation")
Notion of graphs

Degree of a vertex

- **undirected graph:**
  - The **degree** of a vertex = the number of incident edges
  - total # of items in the adj. list = \( \sum_{v \in V} \text{degree}(V) = 2|E| \)

- **directed graph (digraph):**
  - out-degree and in-degree
  - total # of items in the adj. list = \( \sum_{v \in V} \text{out-degree}(V) = |E| \)