Graph

Inst. Nguyễn Minh Huy

Contents

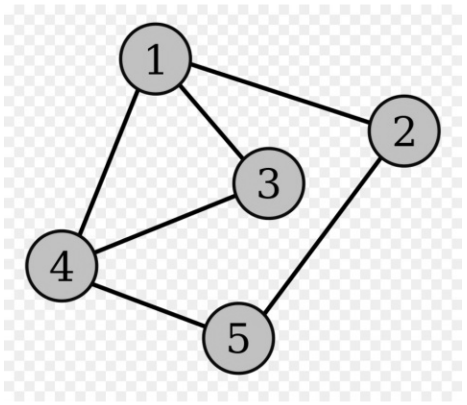
- Graph concepts.
- Implementation.
- Graph visit.

Contents

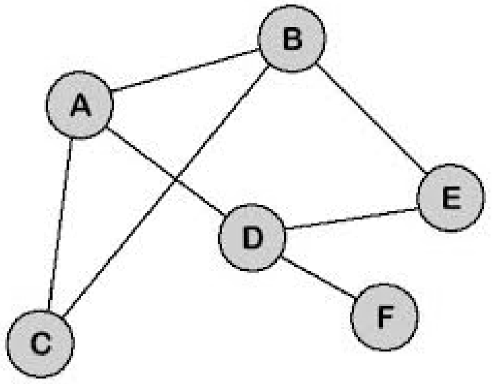
- **Graph concepts.**
- Implementation.
- Graph visit.

Graph concepts

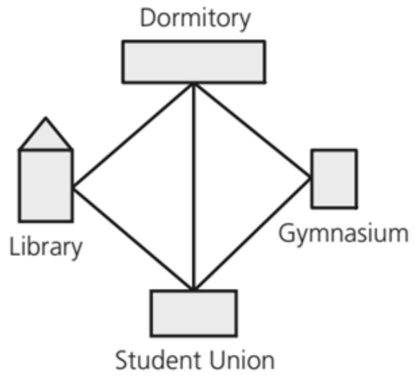
- Definition:
  - Graph G = (V, E), consists of two sets:
    - V: set of vertices, finite, non-empty.
    - E: set of edges, finite.
  - Vertex: node or point in graph.
  - Edge: link or connection between two vertices.



V = { 1, ?? }
E = { (1, 2), ?? }



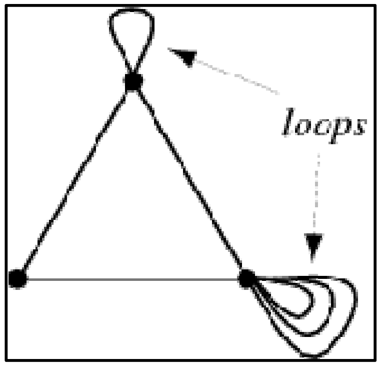
V = { A, ?? }
E = { (A, B), ?? }

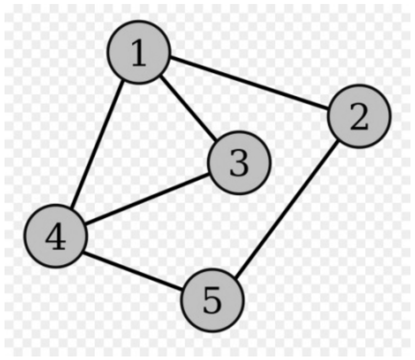


V = { Library, ?? }
E = { (Library, Dormitory), ?? }

Graph concepts

- Basic concepts:
  - Adjacent vertices: two vertices joined by an edge.
  - Path:
    - Series of edges from vertex A to vertex B.
    - Simple path: does not pass a vertex twice.
  - Cycle:
    - Path begins and ends at the same vertex.
    - Simple cycle: does not pass other vertices twice.
    - Loop: self edge, begins and ends at the same vertex.



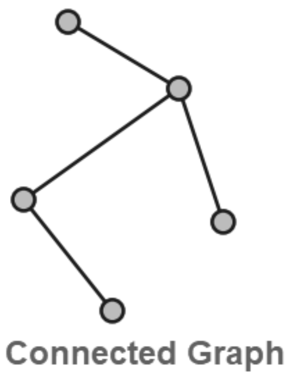


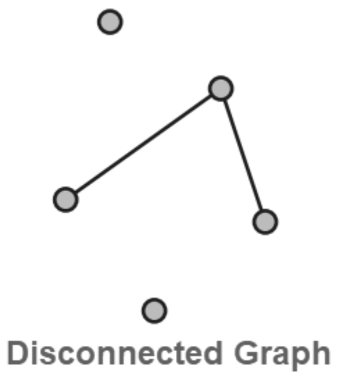
Path(1, 5) = { ?? }

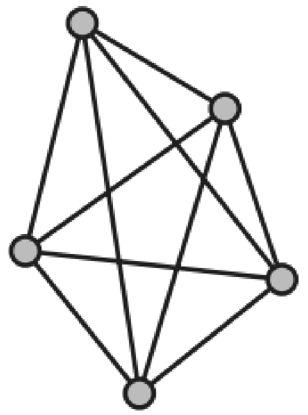
Cycle = { ?? }

Graph concepts

- Graph types:
  - Connected graph:
    - Each pair of distinct vertices has path between them.
    - No isolated vertex.
  - Complete graph:
    - Each pair of distinct vertices has edge between them.
    - Also is a connected graph.

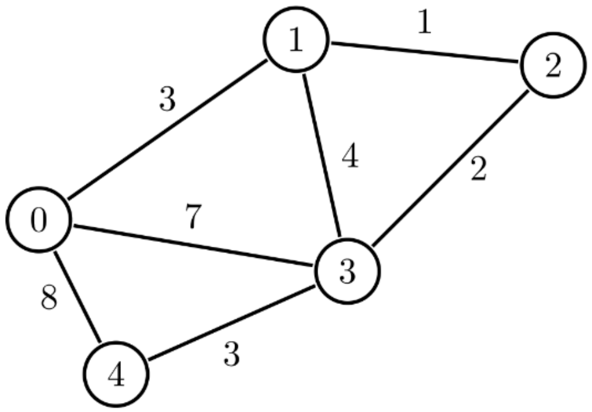

Connected Graph

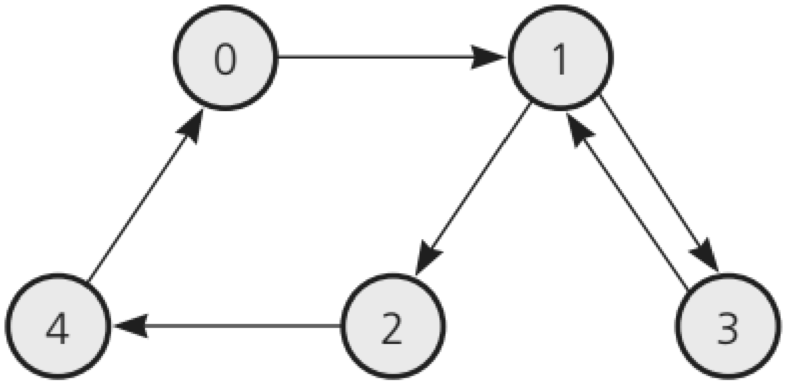

Disconnected Graph



Graph concepts

- Graph types:
  - Weighted graph:
    - Edges are labeled with numeric values.
    - Give meanings to relationships.
  - Directed graph:
    - Edges have direction.
    - Adjacent vertices is not symmetric.



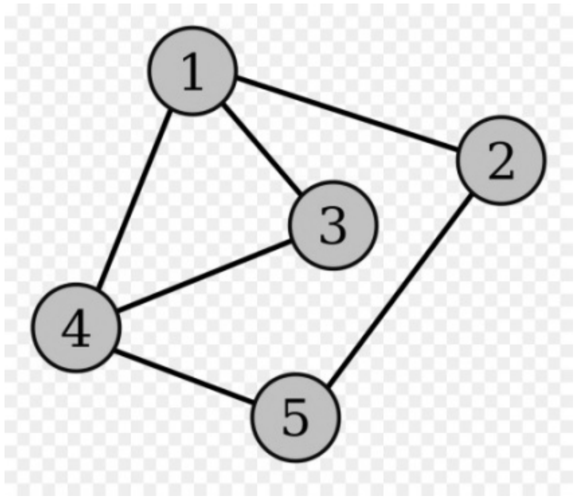


Contents

- Graph concepts.
- **Implementation.**
- Graph visit.

Implementation

- ADT Graph values:
  - Adjacency matrix:
    - N x N matrix for Graph of N vertices.
    - Element (i, j): adjacency between i and j.
      - True/False: has edge or not.
      - Integer/Infinity: has weighted edge or not.

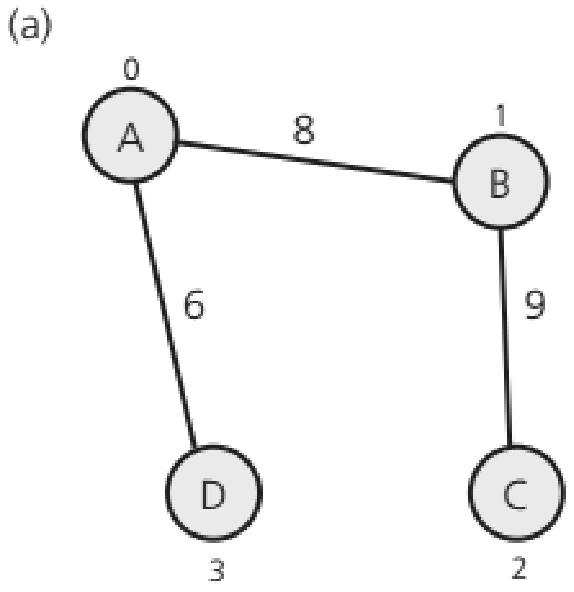


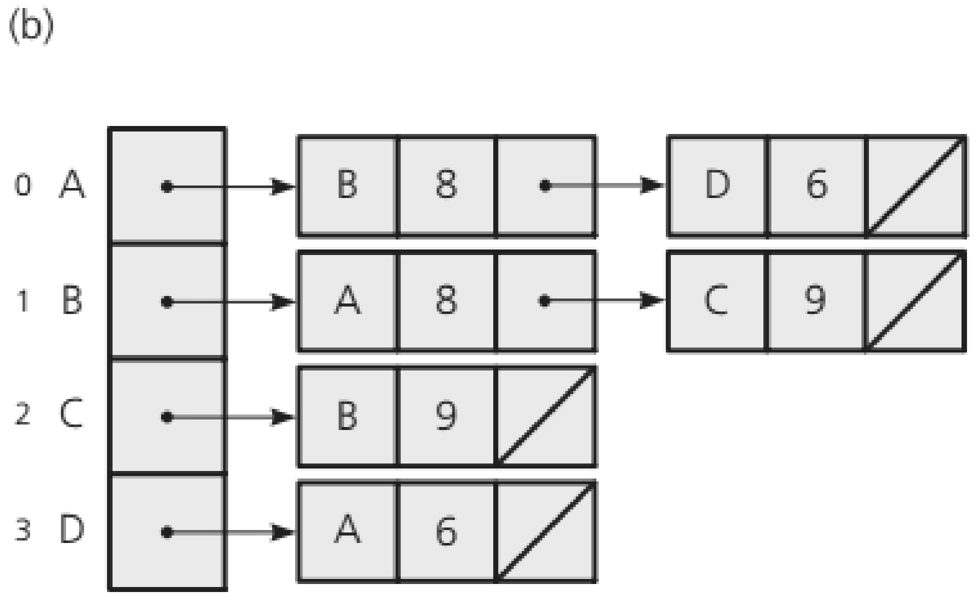
	1	**2**	**3**	**4**	**5**
1	1	1	1	1	0
2	1	1	0	0	1
3	1	0	1	1	0
4	1	0	1	1	1
5	0	1	0	1	1

Implementation

- ADT Graph values:
  - Adjacency list:
    - Array of N singly linked list for Graph of N vertices.
    - If vertex i is adjacent to vertex j:

      ➔ i^{th} linked list contains node j.





Implementation

- ADT Graph operations:
    - Initialize.
    - Check empty.
    - Count vertices.
    - Count edges.
    - Add/remove vertex.
    - Add/remove edge.
    - Tell if two vertices is adjacent.
    - Find all vertices adjacent to a vertex.

Contents

- Graph concepts.
- Implementation.
- **Graph visit.**

Graph visit

- Graphic visit problem:
  - Start from vertex x.
  - Visit all vertex y if there is path.
  - Analysis:
    - Cycle can cause infinite loop!!

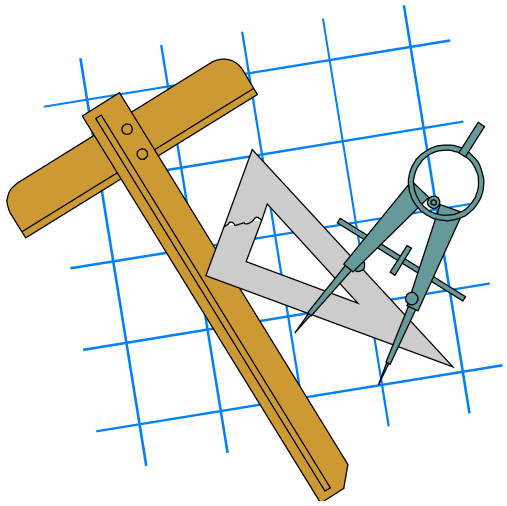
      ➔ Mark visited vertices.
    - Connected component: graph subset of connected vertices.
  - Methods:
    - Depth first search.
    - Breadth first search.

Practice

- Practice 10.1:

  Construct class **Graph** that has the following methods:

  - Initialize.
  - Check empty.
  - Count vertices.
  - Count edges.
  - Add/remove vertex.
  - Add/remove edge.
  - Tell if two vertices is adjacent.
  - Find all vertices adjacent to a vertex.



Practice

- Practice 10.2:

  Provide class **Graph** with the following methods:

  - Print vertices in depth first visit.
  - Print vertices in breadth first visit.
  - Print all connected components.