Undirected Graph Data Type

A screenshot of a graph

Description automatically generated

Representation alternatives

1.Adjacency matrix. An adjacency matrix, where we maintain a V-by-V boolean array, with the entry in row v and column w defined to be true if there is an edge adjacent to both vertex v and vertex w in the graph, and to be false otherwise. Why not? Occupies too much space.

2.Array of edges. An array of edges, using an Edge class with two instance variables of type int. Why not? Implementing adj() would involve examining all the edges in the graph.

3.Array of adjacency lists. An array of adjacency lists, where we maintain a vertex-indexed array of lists of the vertices adjacent to each vertex.

A diagram of a number flow chart

Description automatically generated

Adjacency-list data structure

->Space proportional to V+E

->Constant time to add an edge

->Time proportional to the degree of v to iterate through vertices adjacent to v

A graph with text on it

Description automatically generated with medium confidence

Design pattern for graph processing

The goal is to decouple the implementation from the graph representation. To do so, we develop, for each given task, a task-specific class so that clients can create objects to perform the task.

e.g.

A screenshot of a computer code

Description automatically generated