

Graph

A graph G is a collection of vertices V connected with edges E . Each edge in E is a pair (v_1, v_2) , where v_1 and v_2 are vertices in V .

Directed Graph

A directed graph (digraph) is the graph in which for each edge in E , the ordering of pair (v_1, v_2) matters. The edge (v_1, v_2) and (v_2, v_1) is defined respectively as:

$$V_1 \rightarrow V_2$$

$$V_1 \leftarrow V_2$$

Undirected Graph

An undirected graph is the graph in which for each edge in E , the ordering of pair (v_1, v_2) does not matter. The edge $(v_1, v_2) = (v_2, v_1)$ is defined as:

$$V_1 - V_2$$

Graph Representation

Adjacency Matrix

A graph can be represented using an *adjacency matrix*. For a graph G containing $n = |V|$ number of vertices, its adjacency matrix M of size $n \times n$ is defined as:

$$M_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}$$

For undirected graph, the matrix is symmetric since $(v_1, v_2) = (v_2, v_1)$. This representation allows for constant time search for any edges. However, it takes $O(|V|^2) = O(n^2)$ memory space even when the graph does not have many edges.

Adjacency List

A graph is represented by $|V|$ linked lists for each vertex V storing the outgoing edges. Each edge appears in one of the linked list if the graph is directed or two linked lists if the graph is undirected. Thus in any case, the total size of the graph is $O(|V| + |E|)$.

A more detailed analysis of Graph algorithms is discussed in the **Algorithms** repository.