Theory: Basic terminology

Table of contents:

↑ Basic terminology

§1. Simple graphs

§2. Nodes and edges' relationship

§3. Directed graphs

§4. Paths and cycles

§5. Connectivity in graphs

§6. Subgraphs

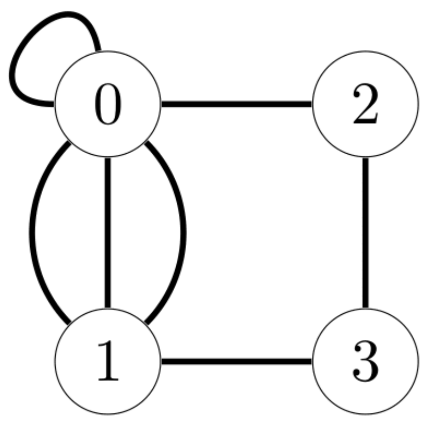
§7. Summary

Feedback & Comments

Previously, we learned some basics about graphs and considered several examples where they might be useful. Also, we discussed that it's quite convenient to represent graphs in a formal way as a pair of two sets: a set of nodes and a set of edges. In this topic, we will continue with graphs and talk about some basic terminology. Although learning terms might seem a bit tedious, this is a very important step. Once you get familiar with basic terms, you will be able to understand more sophisticated algorithms and concepts connected with graphs.

§1. Simple graphs

Up until now, we considered only small graphs with a plain structure. However, there are graphs that look a bit tricky. Look at the following example:



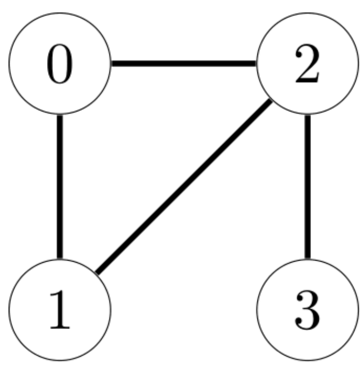
G'

The graph G' consists of four nodes and seven edges. There are two things to note. First: the edge $\{0,0\}$ connects the node 0 with itself. Such an edge is called a **loop**. Also, there are three edges connecting the same pair of nodes: 0 and 1. If a graph contains loops or some edges appear multiple times, the graph is called a **multigraph**. Otherwise, it is called a **simple** graph.

Some objects can be naturally modeled as simple graphs: for example, a political map. In this case, there are no reasons to add loops or multiple edges between any two countries. Now think of a city map: there might be several roads between two points in a city. In this case, it is more appropriate to represent a city as a multigraph.

§2. Nodes and edges' relationship

We already know that graphs consist of nodes and edges: now let's consider in more detail how to describe relations between them. We will use the following example:



G_1

The graph $G_1 = (V_1, E_1)$ consists of four nodes and four edges:

$V_1 = \{0,1,2,3\},\ E_1 = \{\{0,1\},\{0,2\},\{1,2\},\{2,3\}\}$

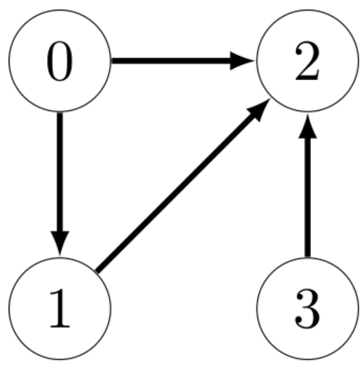
Two nodes of a graph are adjacent if they are connected by an edge. For example, the nodes 0 and 1 in G_1 are adjacent since they are connected by the edge $\{0, 1\}$. If an edge $\{x, y\}$ connects nodes x and y, these nodes are incident to this edge. For example, the node 1 of G_1 is incident to two edges: $\{0, 1\}$ and $\{1, 2\}$. The degree of a node is the number of edges incident to it. For example, the degree of the node 2 of G_1 is equal to 3 since the node is incident to three edges.

To give a real-life interpretation of the described terms, you can consider the nodes of the graph above as people, an edge between two people showing that they know each other. Thus, if two nodes are adjacent, it means that the people that correspond to the nodes know each other. The degree of a node shows how many acquaintances a person has.

§3. Directed graphs

Till this moment, we talked only about graphs where edges don't have any direction. For example, consider the edge $\{0, 1\}$ of G_1 that connects the nodes 0 and 1. If we swap the nodes, it won't change anything: the edge $\{1, 0\}$ still connects the same nodes. Such graphs where the order of nodes is not important are called undirected graphs. A political map is a structure that can be naturally represented as an undirected graph. If one country is a neighbor of another, the direction of the edge showing this relation is not important.

However, in some cases, undirected graphs don't suffice. Assume that we want to model a social network as a graph. We may represent people as nodes and put an edge between two people if one of them follows the other. In this case, the direction is important. If a user x follows y, it doesn't mean that y follows x. To model such a situation, directed graphs suit much better. Below is an example of a directed graph:



G_2

The graph G_2 consists of four nodes and four directed edges:

$V_1 = \{0, 1, 2, 3\},\ E_1 = \{(0, 1), (0, 2), (1, 2), (3, 2)\}$

To show that an edge is directed, we use round brackets instead of curly ones.

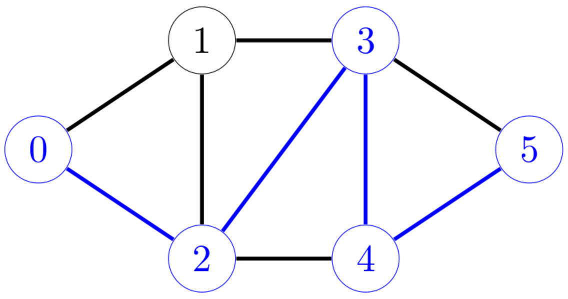
In directed graphs, a node y is said to be adjacent to a node x if there is an edge (x, y) between the nodes. For example, the node 1 of G_2 is adjacent to the node 0 since there is an edge $(0, 1)$. Note that unlike in undirected graphs, the adjacency in directed graphs is not symmetric: if y is adjacent to x, x is not necessarily adjacent to y. As for the term incident, for directed graphs, it remains the same. For example, the nodes 0 and 1 are incident to the edge $(0, 1)$ of G_2.

In directed graphs, each node has an indegree and an outdegree. For a node x, an indegree is the number of nodes incident to x. The number of outgoing edges of x is called an outdegree. For example, an indegree of node 0 in G_2 is 0 and its outdegree is 2.

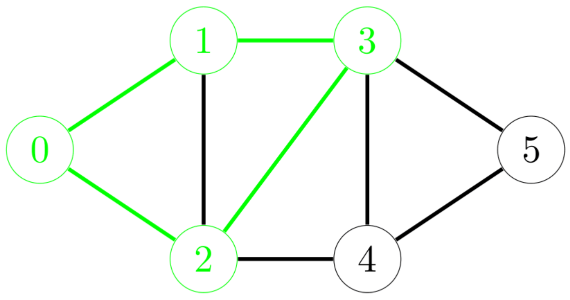
As mentioned before, different types of graphs fit for modeling different problems. Depending on the problem you need to solve, you may use either directed or undirected graphs.

§4. Paths and cycles

A **path** in a graph is an alternating sequence of nodes and edges. Consider the following examples:



G_3

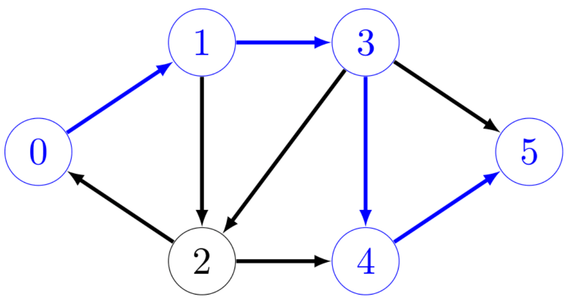


G_3

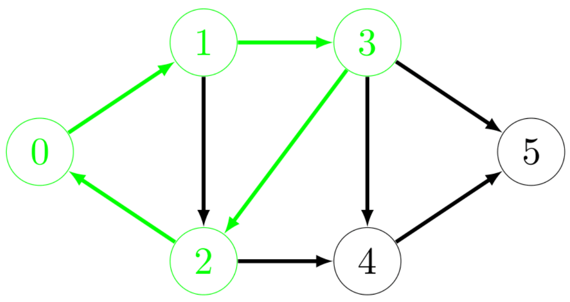
Blue nodes and edges of G_3 (the left figure) correspond to the path $0-2-3-4-5$. Recall the graph that models roads in a city, where edges correspond to roads and nodes correspond to their intersections. Paths are possible ways to get from one point of the city to the other. If each node in a path appears only once, this path is called **simple**. The path above is **a simple path** since there are no nodes appearing multiple times. If we add the edge $\{3, 5\}$ to this path, it won't be simple since now the node 3 appears twice.

If the path starts and ends at the same node, this path is called **a cycle**. For example, green nodes and edges in G_3 (the right figure) form a cycle $0-2-3-1-0$.

For directed graphs, terms are the same. Consider the following examples:



G_4

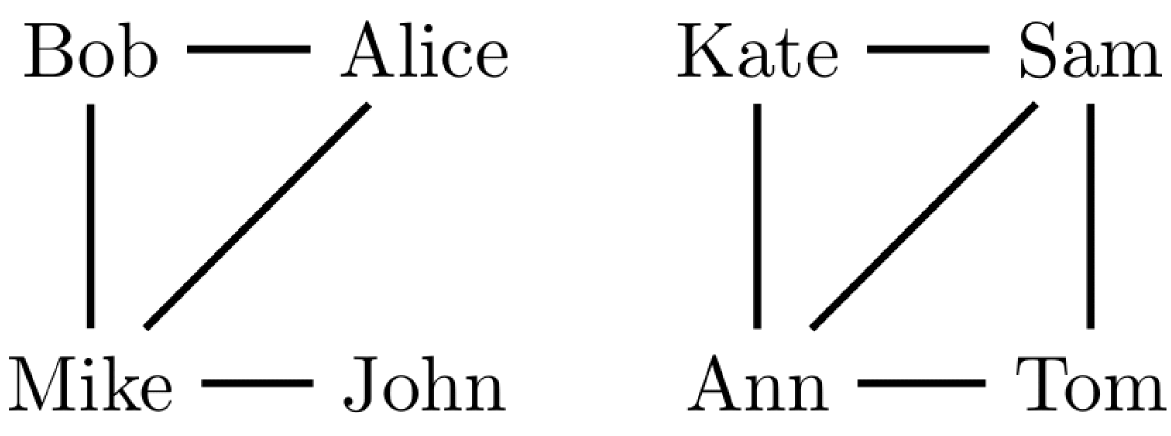


G_4

The left figure shows a path $0-1-3-4-5$ in a directed graph G_4. The right figure shows a cycle $0-1-3-2$ in the same graph. Note that unlike undirected graphs, a path (cycle) from the end to the start node doesn't necessarily exist. For example, there is no path from 5 to 0 in G_4.

§5. Connectivity in graphs

In an undirected graph, two nodes x and y are said to be **connected** if there is a path between the nodes. A set of nodes where each pair is connected by a path forms a **connected component**. Consider the following example:

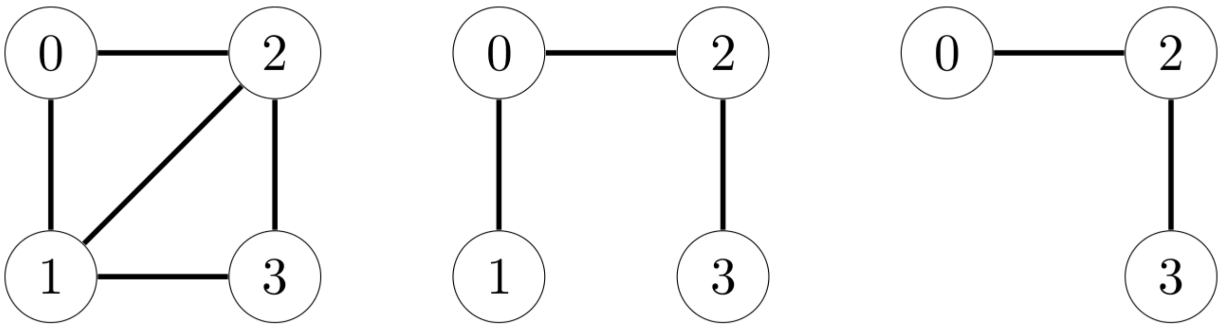


G_5

The graph G_5 represents a group of people, where nodes are connected by an edge if the people know each other. Just looking at this graph, it is easy to understand that it consists of two **connected components**. The first connected component consists of $\{Bob, Mike, Alice, John\}$, and the second one is $\{Kate, Ann, Sam, Tom\}$. A graph G is called a **connected graph** if it contains exactly one connected component. Otherwise, the graph is **disconnected**. Note that now we are talking only about undirected graphs. As for directed graphs, a definition of a connected component is a bit trickier: we will discuss it in the following topics.

§6. Subgraphs

A graph obtained from another graph by removing nodes or edges is called a **subgraph**. Consider the following examples:



G_6 G_7 G_8

Here, the graph G_7 is a subgraph of G_6 because it was obtained from G_6 by removing the edges $\{1, 3\}$ and $\{1, 2\}$. G_8 is a subgraph of G_6 as well since it was obtained from G_6 by removing the node 1. Note that when we remove a node from a graph, we also remove all edges incident to this node.

To give a real example, assume that each node of G_6 corresponds to a computer network, and edges show computers that are connected to each other. A question that we might ask here is what's the minimum number of links needed to keep all the computers connected? The subgraph G_7 shows one of the possible answers: if we remove at least one edge, the computers become disconnected.

§7. Summary

In this topic, we considered the basic terminology used for graphs. We learned that there are undirected and directed graphs and considered relations between nodes and edges in more detail. Also, we touched on paths, connectivity, and subgraphs. Although this is only the beginning, this vocab is already enough to discuss more sophisticated graph algorithms and concepts.

Report a typo

Feedback sent!

Start practicing

Comments (17) Hints (0) Useful links (0) Show discussion