Theory: Connectivity in graphs

12 minutes 7 / 7 problems solved

Start practicing

For now, we've already learned what a graph is and discussed some basic terminology. One more important concept that we should cover is **connectivity**. This concept is often used in graph theory and arises in different graph algorithms, so it is important to become familiar with it. In this topic, we will discuss this concept in detail: first, for undirected graphs and then for directed ones.

Current topic:

Connectivity in graphs

Topic depends on:

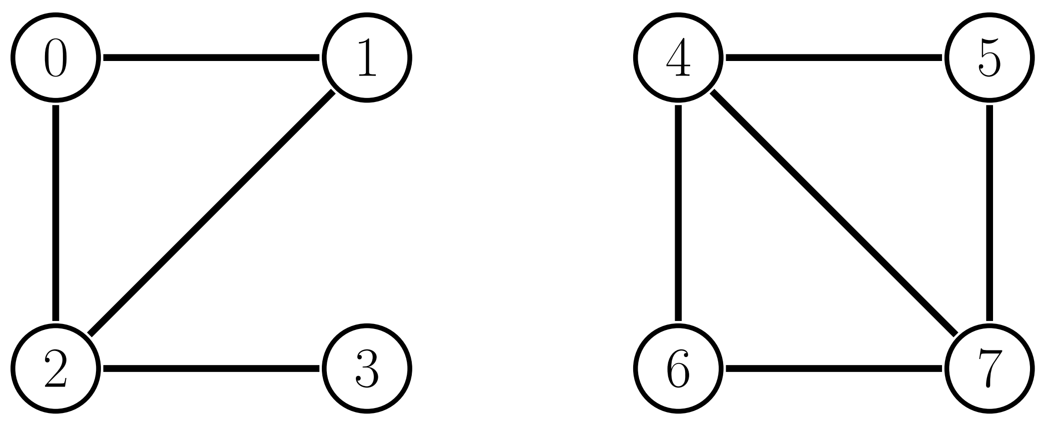
Basic terminology

Topic is required for:

Kosaraju's algorithm

§1. Connectivity in undirected graphs

In an undirected graph, two nodes are said to be **connected** if there exists a path between these nodes. For example, the nodes 1 and 3 of G_1 are connected since there is a path $1-2-3$:



G_1

An example of nodes that are not connected is 3 and 6: as you can see there is no path that's connecting them.

A subgraph of a graph is called a **connected component** if

- each pair of nodes in this subgraph is connected;
- no other nodes can be added to this component without breaking its property of being connected.

According to this definition, G_1 consists of two connected components. The first one includes the nodes 0, 1, 2 and 3, since

- there is a path between each pair of these nodes;
- if we add one more node from G_1 to this subgraph, it will not be connected.

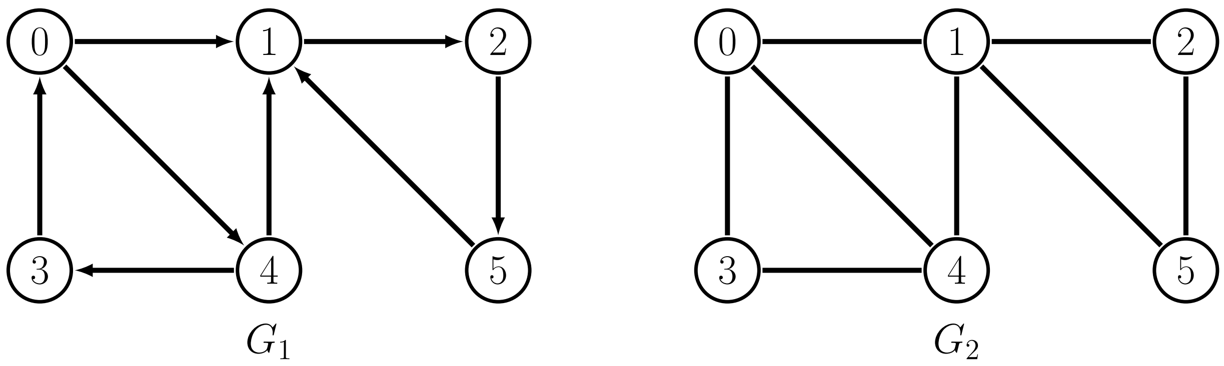
The second connected component contains the nodes 4, 5, 6, and 7.

A graph G is called **connected** if it consists of only one connected component. As you can see, G_1 does not fit this definition, thus it's **disconnected**.

§2. Directed graphs: weak connectivity

As for directed graphs, the concept of connectivity is a bit more complex than for undirected ones. There are two different types of connectivity for directed graphs: **weak** and **strong**.

A directed graph is **weakly connected** if there exists an **undirected path** between each pair of nodes in the graph. In other words, if we transform a directed graph into an undirected by removing the direction of each edge and get a connected graph, then the directed graph is weakly connected. Let's consider the following example:



If we remove the direction of each edge of the graph G_1 we will get a connected undirected graph G_2. Therefore, G_1 is weakly connected.

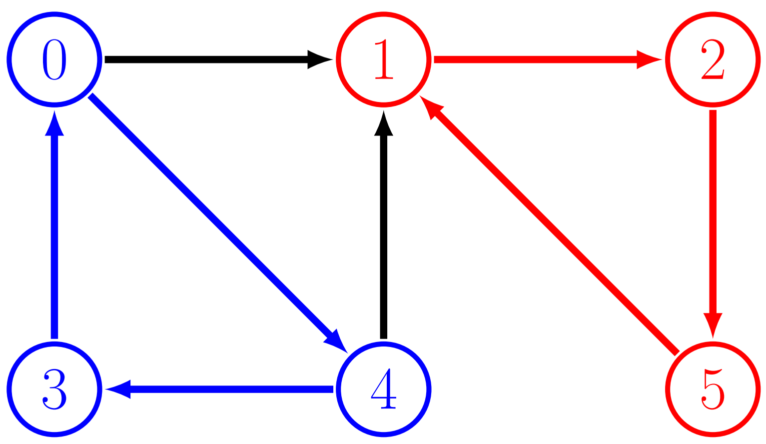
§3. Directed graphs: strong connectivity

A directed graph is **strongly connected** if there exists a **directed path** between each pair of nodes in the graph. According to this definition, G_1 is not strongly connected. For example, there is a path $3 \to 0 \to 1 \to 2$ between the nodes 3 and 2, but there is no path between 2 and 3.

A subgraph of a directed graph is a **strongly connected component** (shortly, SCC) if:

- there exists a directed path between each pair of nodes in this subgraph;
- no other nodes or edges can be added to this subgraph without breaking its property of being strongly connected.

The graph G_1, for example, consists of two strongly connected components:



The blue nodes 0, 3, and 4 belong to the first component, while the red nodes 1, 2, and 5 represent the second one. If we merge the nodes of each strongly connected component into one, we will get a directed acyclic graph, a **condensation** of G_1:

§4. Summary

In this topic, we've learned about connectivity in undirected and directed graphs. Keep in mind that unlike undirected graphs, directed ones have two types of connectivity: weak and strong. Intuitively, each connected component in a graph represents a set of nodes that are "more closely related" to each other compared to the nodes belonging to different components. For example, If we model a social network using a directed graph, two people from one strongly connected component are more likely to know each other. Thus, one of them might be presented to another as a friend.

Table of contents:

Start practicing

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