Topological Sort: Introduction

Let's go over the Topological Sort pattern, its real-world applications and some problems we can solve with it.

We'll cover the following

Overview

The **topological sort** pattern is used to find valid orderings of elements that have dependencies on, or priority over each other. Scheduling and grouping problems that have prerequisites or dependencies generally fall under this pattern.

In simple terms, topological order can be described as a partial order. In comparison to a total order, it offers more flexibility since there can be multiple correct answers. Consider sorting an array $[5, 2, 1]$ in ascending order. This gives a unique result $[1, 2, 5]$, which is the only correct answer—a complete order. On the other hand, consider an example of course prerequisites where English I has to be taken before English II, and Mathematics can be taken whenever. Since there's no obligation on when to take Mathematics, we can take it in any order as long as the dependency is respected—that is, as long as English I is taken before

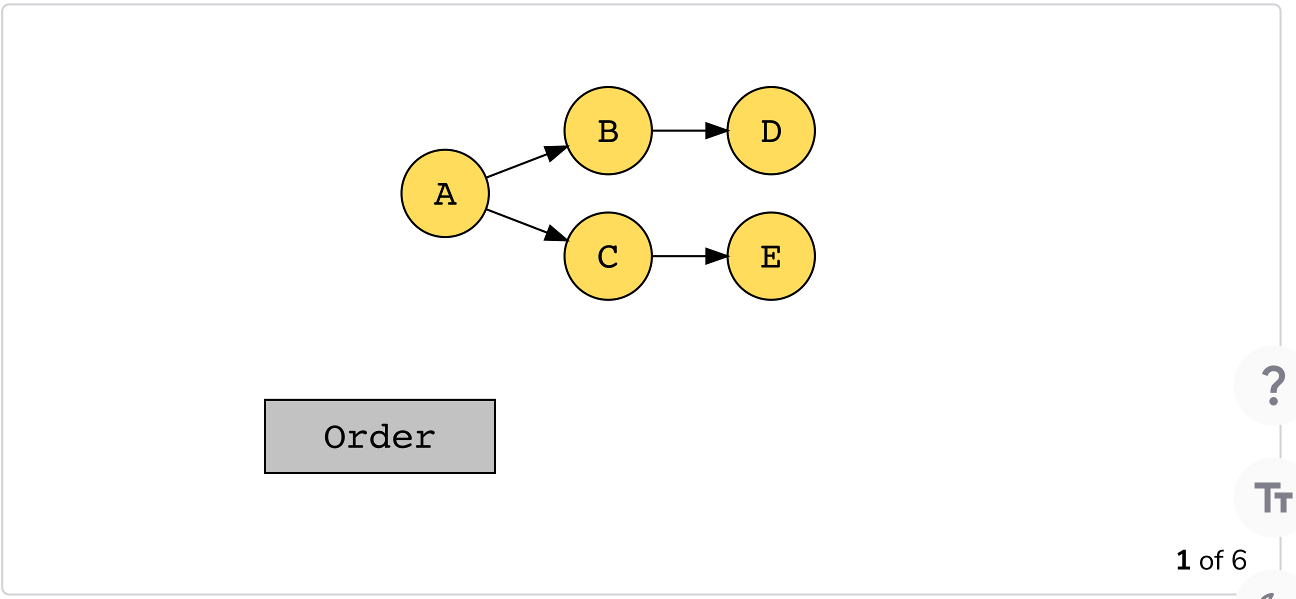
English II. This is an example of partial order, or topological order, since there are three valid orders in which a student may take these courses:

- English I, English II, Mathematics
- Mathematics, English I, English II
- English I, Mathematics, English II

With the topological sort pattern, converting a problem to a directed graph is the first step to finding a solution. The elements to consider correspond to graph vertices and the dependency relationships form the edges. A valid topological order exists if the graph is acyclic. For example, if B is dependent on A, there is an edge from A to B, hence, A comes before B in a linear ordering. However, if an edge exists from B to A as well, that is, there is a cycle, a linear ordering is not possible.

Consider another example where an edge exists from vertex A to vertex B and vertex C. In this case, after adding A to our ordering, there are two possible options, B or C. Both of them are valid, so we get two orders: $[A, B, C]$ or $[A, C, B]$.

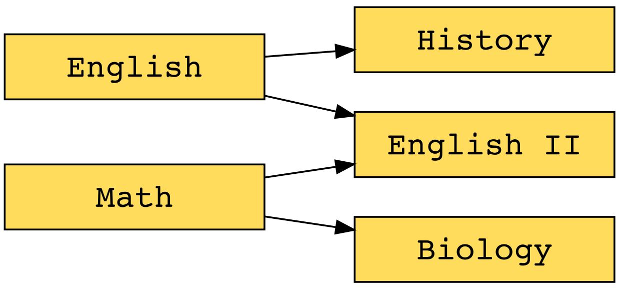
The following illustration shows a simple example of finding the topological order from a graph:



Examples

The following examples illustrate some problems that can be solved with this approach:

Course prerequisites



Two of many possible orderings:

English	Math	History	English II	Biology
English	History	Math	English II	Biology

1 of 2

Does my problem match this pattern?

- Yes, if either of these conditions is fulfilled:
    - A partial ordering of the given elements is required.
    - The problem asks you to find an ordering of elements based on some dependency rules between them.
- No, if either of these conditions is fulfilled:
    - A total ordering of the given elements is required.
    - The problem relationships cannot be converted to a graph.
    - The corresponding graph is not a tree.

Real-world problems

Many problems in the real world use the topological sort pattern. Let's look at an example.

Topological sort can be used for process scheduling in a computer system. When a system is booted, the operating system needs to run a number of processes. Some processes have dependencies, which are specified using ordered pairs of the form (a, b). This means that process b must be run before process a. Some processes don't have any dependencies, meaning they don't have to wait for any processes to finish. Additionally, there cannot be any circular dependencies between the processes like $(a, b)(b, a)$. In order to successfully start the system, the operating system needs to select an order to run the processes in. The processes should be ordered in such a way that whenever a process is scheduled to run, all of its dependencies are already met.

These processes can be represented in the form of a graph where the vertices are the processes and the edges represent their dependency

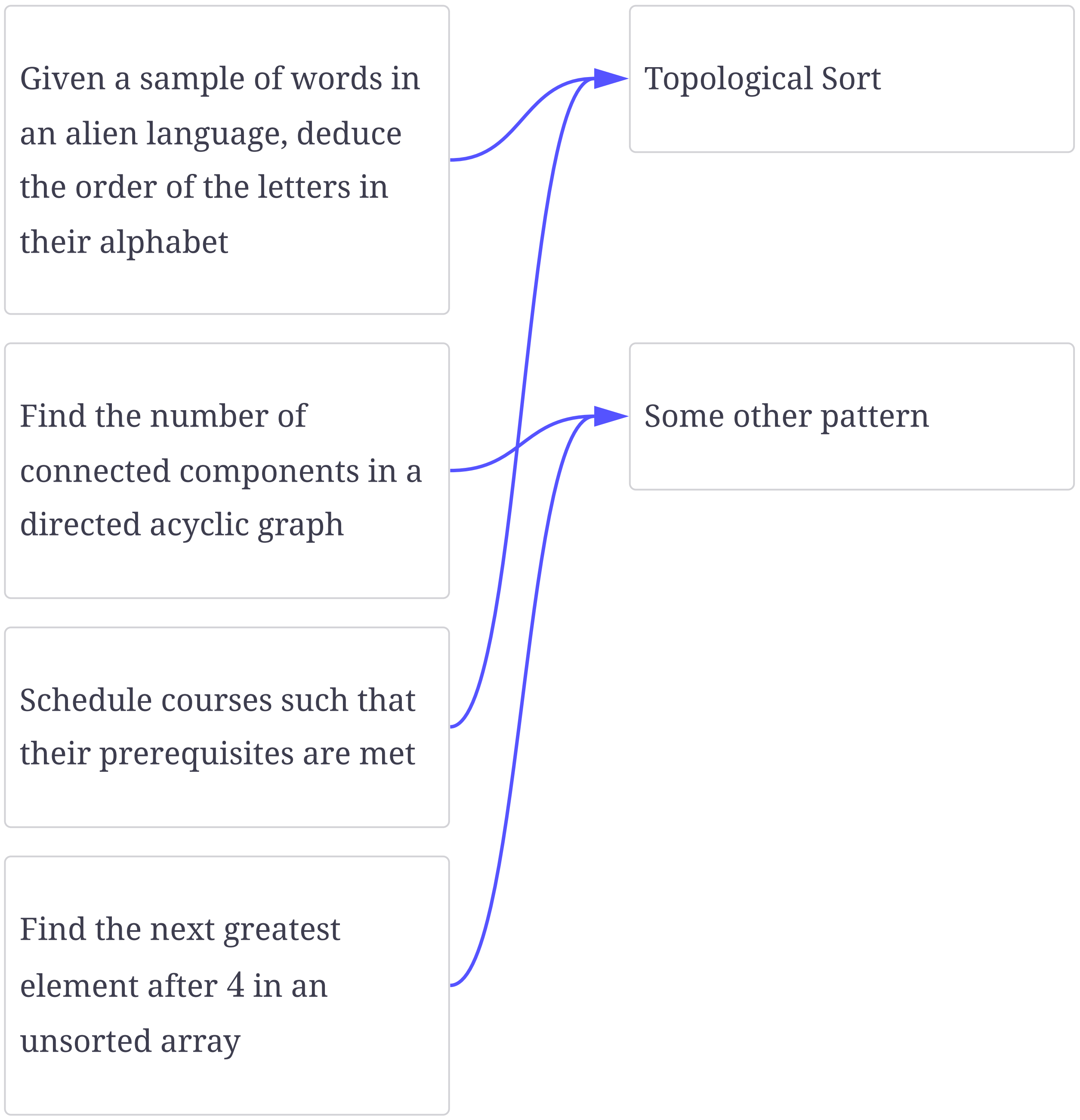
Strategy time!

Match the problems that can be solved using the topological sort pattern.

> **Note**: Select a problem in the left-hand column by clicking it, and then click one of the two options in the right-hand column.

Match The Answer

ⓘ Select an option from the left-hand side



Given a sample of words in an alien language, deduce the order of the letters in their alphabet

Find the number of connected components in a directed acyclic graph

Schedule courses such that their prerequisites are met

Find the next greatest element after 4 in an unsorted array

Topological Sort

Some other pattern

Reset Show Solution Submit

← Back

Find the First K Missin...

Next →

Compilation Or

?

✓ Mark as Completed

Tт