Compilation Order

Try to solve the Compilation Order problem.

We'll cover the following

Statement

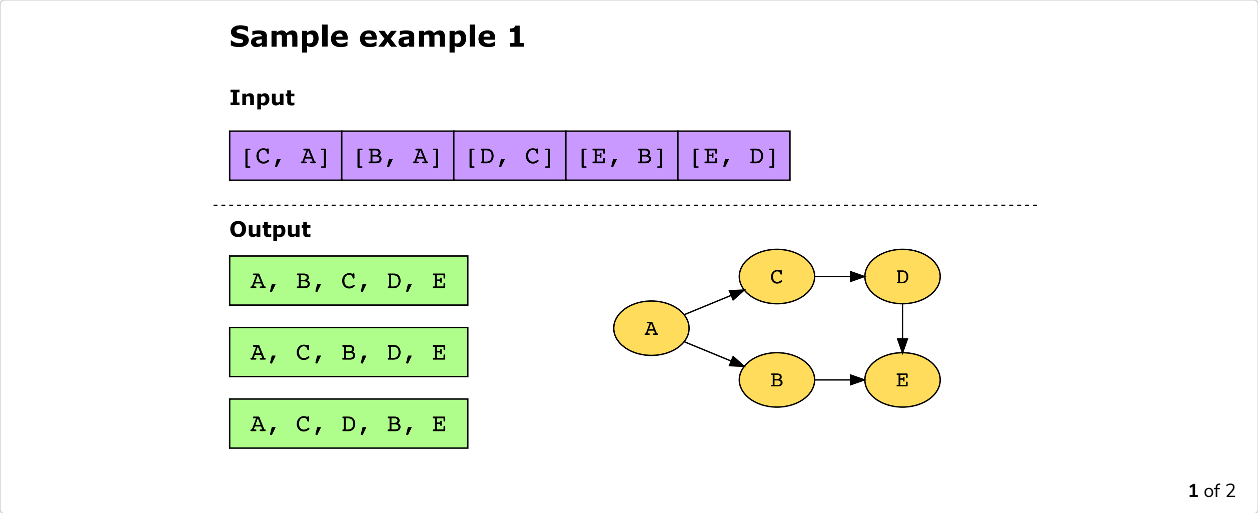
There are a total of n classes labeled with the English alphabet (A, B, C, and so on). Some classes are dependent on other classes for compilation. For example, if class B extends class A, then B has a dependency on A. Therefore, A must be compiled before B.

Given a list of the dependency pairs, find the order in which the classes should be compiled.

Constraints:

- Class name should be a character.
- $0 \leq$ `dependencies.length` ≤ 5000
- `dependencies[i].length` $= 2$
- All dependency pairs should be unique.

Examples



Sample example 1

Input

[C, A] [B, A] [D, C] [E, B] [E, D]

Output

A, B, C, D, E

A, C, B, D, E

A, C, D, B, E

1 of 2

In the above examples, the arrows represent the relationship between these classes. For example, the arrow $A \rightarrow C$ shows that C extends A, and therefore, C is dependent on A.

Understand the problem

Let’s take a moment to make sure you’ve correctly understood the problem. The quiz below helps us to check if you’re solving the correct problem:

Compilation Order

1

Given the following list of dependencies, what is the order of compilation of classes?

Select all that apply.

dependencies = [A, B], [B, C], [A, D]

□ A) [A, B, C, D]

□ B) [C, A, B, D]

□ C) [C, B, D, A]

□ D) [D, C, B, A]

Submit Answer

‹ Question 1 of 3 ›
0 attempted

Reset Quiz

Figure it out!

We have a game for you to play. Rearrange the logical building blocks to develop a clearer understanding of how to solve this problem.

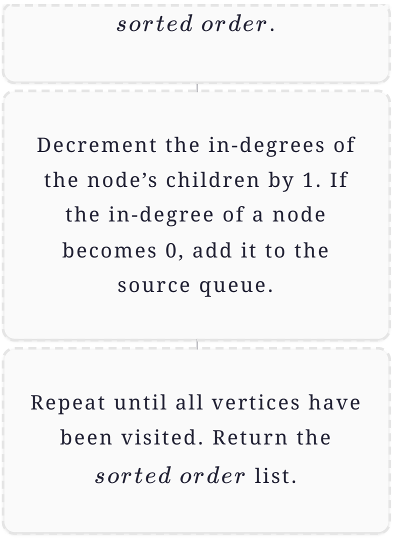
Drag and drop the cards to rearrange them in the correct sequence.

Build the graph from the input using adjacency lists.

Store the in-degree of each vertex in a hash map.

Add the sources to a queue.

Pop from the queue and store the node in a list, let’s call it



sorted order.

Decrement the in-degrees of the node's children by 1. If the in-degree of a node becomes 0, add it to the source queue.

Repeat until all vertices have been visited. Return the *sorted order* list.

Reset | Show Solution | Submit

Try it yourself

Java

usercode > CompilationOrder.java

```java
import java.util.*;

class CompilationOrder {

    public static List<Character> findCompilationOrder(ArrayList<ArrayList<Character>> dependencies){
        // Write your code here
        return new ArrayList<Character>();
    }
}
```

Powered by AI | Submit

Test Cases | Results

Case 1 | Case 2 | Case 3

Input #1

[["B","A"],["C","A"],["D","C"],["E","D"],["E","B"]]

Compilation Order

← Back
Topological Sort: Intro...

Next →
Solution: Compilation ...

☑ Mark as Completed