

Intro to Algorithms

Homework

Topological Sort: Alternative to DFS Approach For Topological Sort

Sedgewick has the following tantalizing statement about topological sort (page 583).

Despite the simplicity of this algorithm, it escaped attention for many years, in favor a more intuitive algorithm based on maintaining a queue of vertices of indegree 0.

The Assignment

Your assignment is to implement a *Queue based topological sort* (Sedgewick 4.2.39 on page 601).

You may use any JDK class such as `java.util.{Set,Queue}` etc.

You may use any Sedgewick code from the textbook.

You **must** implement the *Digraph* API from Sedgewick (page 569) as a class named `Digraph.java`, copying his code if you want.

You **must** implement the topological sort code in a class named `TopologicalQ.java` and provide the following public API

- Constructor, `TopologicalQ(Digraph G)`. The constructor determines whether the digraph has a cycle or not. If it does not have a cycle, the ctor determines a valid topological sort for the digraph.
- `boolean hasOrder()`: return true iff G has a topological order, false otherwise
- `boolean Iterable<Integer> order()`: return an Iterator over the vertices in a valid topological sort. The method returns `null` if no topological order exists.

This style of code should be familiar to you from the textbook.

Packaging

Assume that you've installed your YU Git repo in a directory named `GIT`.

- Your homework assignments for this course must be rooted in `GIT/IntroToAlgorithms/homework`.

I'll refer to this directory as `ROOT` .

- Your code will reside in a package named `edu.yc.oats.algs` .
- Your code for this assignment will be rooted in: `ROOT/TopologicalQ` . I'll refer to this directory as `DIR` .
- Your classes must reside `DIR/src/main/java/edu/yc/oats/algs` .
- Your assignment may not use any external libraries: just the SDK and your code in this package.

Grading

- If your code cannot be compiled & run (either because it doesn't follow the packaging conventions above, or because of a compilation bug) -- **automatic 0** for the assignment
- If your code runs, but doesn't pass my tests, you'll get a *maximum* of 8. The actual grade will depend on how close your code was to passing the tests.
- If your code runs, passes the tests, but is "really ugly", you get a 9.
- Maximum grade is 10.