# DAG Serialization

*Problem author: Vitaly Aksenov; problem developer: Dmitry Petrov*

Consider the lifetime of our register. Notice that it has no more than three phases:

1. The initial "`false`" phase
2. The intermediate "`true`" phase
3. The final "`false`" phase

Additionally, there are no more than 4 types of operations:

1. "`set true`"
2. "`unset true`"
3. "`set false`"
4. "`unset false`"

The operation of the 1st type "`set true`" transitions between the 1st and the 2nd register phases. The operation of the 2nd type "`unset true`" transitions between the 2nd and the 3rd register phases.

All the operations of the 3rd type are between the 1st and the 2nd. We can add logical arcs from the operations of type 1 to all the operations of type 3, and from all the operations of type 3 to the operation of type 2. Also, add a logical arc from the 1st to the 2nd operation.

Operations of the 4th type may occur in either the 1st or 3rd phase. Their exact placement can only be determined by the DAG.

We can finally build a graph $G$ containing $n$ vertices (all the operations), $m$ given DAG arcs, and the logical arcs mentioned. Then, simply find a topological sort in $G$, starting from the operation of type 1, ensuring there are no loops, and return it as the answer.