Graph Theory: Topological Sorting

David Jacobo
jguillen@cimat.mx

October 29, 2015

Definition

$$G = (V, E)$$

Definition

Indegree: Basically tells us how many edges points to each of the vertices, (u, v) augments v indegree in 1.

Outline

Intuition

Given a directed-graph (aka **digraph**) in which the **vertices are tasks** and **edges represent dependencies between them**; tell if there is a permutation of the vertices which satisfies all the dependencies.

Algorithm

Algorithm 1 Compute indegree

1: $\forall\, u \in V, u_{indegree} \leftarrow 0$
2: **for** $u \in V$ **do**
3: **for** $v \in V$ **do**
4: **if** $(u, v) \in G$ **then**
5: $v_{indegree} \leftarrow v_{indegree} + 1$
6: **end if**
7: **end for**
8: **end for**

Algorithm

Algorithm 2 Set initial vertices

1: **for** $u \in V$ **do**
2: **if** $u_{indegree} == 0$ **then**
3: $output \leftarrow output + u$
4: **end if**
5: **end for**

Algorithm

Algorithm 3 Top-sort

```
1: Compute indegree
2: Set initial vertices
3: for u ∈ output do
4:     for v ∈ V do
5:         if (u, v) ∈ G then
6:             v_indegree ← v_indegree − 1
7:             if v_indegree == 0 then
8:                 output ← output + v
9:             end if
10:        end if
11:    end for
12: end for
```

Keep in mind...

- Remember the **indegree** concept.
- Generalize the algorithm as a way to sort tasks in a linear way.
- Never try to memorize code, but rather implement your own version and stick with it :)
- Customize (Pimp?) your code based on the problem.

Q & A

References

- Competitive Programming site
- Algorists' repository