Graph Algorithms

Jayati Kaushik

St. Joseph's University, Bengaluru

Foundations of Data Science
BDA2121

Algorithm for computing $E_G(s)$

Input: A Digraph G in adjacency list representation, a vertex $s \in V(G)$; a marking number $p \in \mathbb{N}$

Set mark[s]:= p and mark[v] := nil $\forall v \in V \setminus \{s\}$
L := (s)
while L $\neq \emptyset$ **do**
    Delete the first element u from L.
    for all $v \in$Adj[u] **do**
        if mark[v] = nil **then**
            Set mark[v] := p and add v to the end of L.
        end if
    end for
end while
return $E_G(s) := \{v \in V :$ mark[v]=p$\}$

Acyclic Graph

Definition

A graph is called *acyclic* if it has no simple cycle.

Topological Sorting

Defition

A *topological sorting* of a directed graph $G = (V, R, \alpha, \omega)$ is a bijection $\sigma : V \to \{1, 2 \ldots, n\}$ such that $\forall r \in R$

$$\sigma(\alpha(r)) < \sigma(\omega(r))$$

Condition for Acyclic

Theorem

G has a topological sorting iff G is acyclic.

Proof $\Leftarrow$

If G has a cycle $C = (v_0, \ldots, v_k)$ then G cannot have a topological sorting since then:

$$\sigma(v_0) < \sigma(v_1) < \ldots \sigma(v_k) = \sigma(v_0)$$

Topological Sorting

Proof $\Rightarrow$

The proof is by induction. For, $n = 1$ it is trivial. Let $n > 1$.
Let G be acyclic.
Then there is atleast one $\tilde{v} \in V$ with no outgoing edges.
By induction, $G - \tilde{v}$ has a topological sorting σ'. For $v \in V$ we set

$$\sigma(v) = \begin{cases} 1 & v = \tilde{v} \\ \sigma'(v) + 1 & v \neq \tilde{v} \end{cases}$$

Algorithm for Topological Sort

Input: A Digraph in adjacency list representation.

1. Compute the indegrees igrad[v] for all $v \in V$.
2. Set $L_0 := \{v \in V : \text{igrad}[v] = 0\}$.
3. **for** $i = 1, \ldots, n$ **do**
4.   Delete the first vertex v_i from L_0 and set $\sigma(v_i) = i$.
5.   **for all** $r \in \delta^+(v_i)$ **do**
6.   Let $w = \omega(r)$ the endvertex of r, igrad[w] := igrad[w]-1.
7.   **if** igrad[w] $= 0$ **then**
8.   Add w to L_0
9.   **end if**
10.   **end for**
11. **end for**
12. **return** σ

Bipartite Graphs

Definition

A graph G is *bipartite* if there is a partition $V(G) = A\dot{\bigcup}B$ such that every arc had one end vertex in A and the other in B.

Theorem

An undirected graph G is bipartite iff G has no cycle of odd length.