Algorithms, Design & Analysis

Lecture 12: **Articulation points and Topological sort**

Muhammad Ibrahim & Rana Hammad Ahmad

Information Technology University

June 19, 2025

About Your Fellows

- Hi there! We are **Ibrahim** and **Hammad**.
- We are Associate Students at ITU.

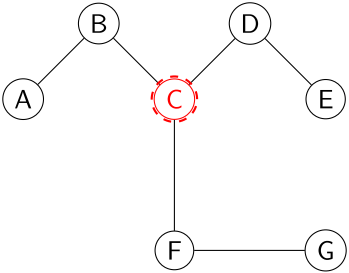
Articulation Points

Articulation Points

Definition:

- Articulation point is a node which if removed increase the number of connected components.

Articulation Point Example



Explanation: The vertex C is an articulation point because its removal disconnects the graph into three separate components.

Algorithm For Articulation Points

Articulation Points Algorithm

Algorithm 1 Finding Articulation Points

for each $x \in V$ **do**
    $flag \leftarrow$ False
    $t \leftarrow 1$
end for
for each $v \in N(x)$ **do**
    if v is unvisited **then**
        DFS(v)
    end if
    if $flag =$ True **then**
        v is an **Articulation Point (ARTP)**
    end if
    $flag \leftarrow$ True
end for

Depth-First Search Algorithm For Articulation Points

DFS Algorithm

Algorithm 2 DFS(v)

$v.d = t, v.l = t, t \leftarrow t + 1$
$v.visited \leftarrow$ True
for each $u \in N(v)$ **do**
    if u is unvisited **then**
        DFS(u)
    end if
    if $u.d < v.d$ and $u \neq v.\pi$ **then**
        $v.l \leftarrow \min(u.d, v.l)$
    end if
    if $u \neq v.\pi$ **then**
        $v.l \leftarrow \min(v.l, u.l)$ checking if their is a path to ancestors via child
    end if
    if $u.l > v.d$ **then**
        v is an **ARTP**
    end if
end for

Complexity

- Runtime complexity of this algorithm is $\Theta(|V| + |E|)$.

Topological Sort

Topological Sort

Definition:

- It is linear ordering of graph vertices such that for every directed edge uv from vertex u to vertex v,u comes before v in the ordering.

Topological Sort

Conditions:

- Tree: Minimally connected graph (there exist a path between every pair vertices).
- Directed Acyclic Graph (DAG)
- It checks indegrees and outdegrees

Topological Sort

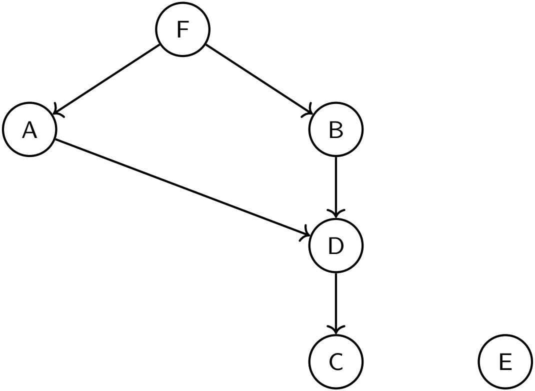
Example:

- **Task Scheduler**
  - A Task Scheduler is a system that manages the execution order of tasks while considering dependencies. This is a perfect real-world application of Topological Sorting, which is used in Directed Acyclic Graphs (DAGs) to order tasks such that each task appears before any tasks that depend on it.

How Topological Sorting Works in a Task Scheduler

- **Tasks as Graph Nodes:**
  - Each task is represented as a **node** in a directed graph.
- **Dependencies as Directed Edges:**
  - A directed edge $A \rightarrow B$ means **task A must be completed before task B**.
- Check if in-degree is zero. If it is remove it and put it in the list.

Task Dependency Graph (Topological Sorting)



Valid Execution Order: F → A → B → D → C → E

Topological Sorting Algorithm (1/2)

Algorithm 3 Topological Sorting (TS)

```
Given a directed graph G = (V, E)
Compute in-degrees of all vertices
Initialize an empty queue W ← [ ]
for each u ∈ V do
    if d(u) = 0 then
        Add u to W
    end if
end for
```

Topological Sorting Algorithm (2/2)

Algorithm 4 Topological Sorting (TS) - Continued

while W is not empty **do**
    Remove a vertex u from W
    for each outgoing edge (u, v) **do**
        Remove edge (u, v) from G
        Decrease in-degree of v
        if $d(v) = 0$ **then**
            Add v to W
        end if
    end for
end while

Topological Sort(Home Work)

- Prove this Topological Sort in Linear time $O(V + E)$.