Structure Theorem for Directed Graphs

Definition: A directed graph $G(V,E)$ is called **acyclic** if it has no directed cycle.

**Theorem #1:** A directed graph $G(V,E)$ is acyclic if and only if its vertex set $V$ can be labeled/ordered as $V = \{1, 2, \ldots, n\}$ so that

$$\forall (i,j) \in E, \quad i < j.$$ 

(i.e. its vertices can be ordered on a line so that all edges go forward.)

**Theorem #2:** For an acyclic graph $G(V,E)$, the ordering as in Theorem #1 can be computed in $O(m+n)$ time $m = |E|, n = |V|$ given the adjacency list representation. Such an ordering is called **topological ordering**.
Def A directed graph $G(V,E)$ is called strongly connected if $\forall u,v \in V$, there is a directed $u \rightarrow v$ path (and $v \rightarrow u$ path).

Fact Given adjacency list representation, one can check if $G(V,E)$ is strongly connected in $O(m+n)$ time. (Using BFS)

Structure Theorem (Informally)

Every directed graph $G(V,E)$ can be decomposed into strongly connected components - along with, acyclic (topological ordering) structure imposed on these components.

$G(V,E)$:

\[ C_1 \rightarrow \ldots \rightarrow C_i \ldots \rightarrow C_j \rightarrow \ldots \rightarrow C_k \]
Theorem #3
For every directed graph $G(V, E)$, we can decompose $V$ as $V = C_1 \cup C_2 \cup C_3 \ldots \cup C_k$ such that
- Every $G'_C_i$ is strongly connected ($G'_C_i$ is restriction of vertices and edges to $C_i$).
- For all other edges $(a, b) \in E$, $a \in C_i$, $b \in C_j$ for some $i < j$.

Theorem #4 The decomposition into strongly connected components as in Theorem #3 can be computed in $O(m+n)$ time given adjacency list representation.
Strategy to design algorithms on directed graphs

1. Design algorithm (in the special case) when the graph is acyclic. (often dynamic pgm.)

2. On a general graph, first find its decomposition into strongly connected components as in Theorem #3, 4.

3. And then combine / generalize Algorithm in step 1 with / using the decomposition in step 2.