4 types of edges:

- **Tree edge**: edge $v \rightarrow u$, explore $(u)$ is called as a recursive call within explore $(v)$.
- **Forward edge**: edge $v \rightarrow u$, where in the (solid) tree:
  - $u$ in subtree of $v$,
  - $u$ is not a child of $v$.
  - in the figure: $(A,F), (E,G)$
- **Back edge**: edge $v \rightarrow u$, where in the (solid) tree:
  - $v$ in subtree of $u$.
  - in the figure: $(F,B), (D,A)$
- **Cross edge**: all other edges.
  - in the figure: $(D,H), (H,G)$
How to decide the type of an edge?

**Tree edges**: these are discovered during the algorithm.

Observe that for tree edge \((v,u)\): \(\text{pre}(v) < \text{pre}(u) < \text{post}(u) < \text{post}(v)\).

**Forward edges**: edge \((v,u)\) with

\[(u,v) \neq \text{tree edge}\]

and

\[\text{pre}(v) < \text{pre}(u) < \text{post}(u) < \text{post}(v)\]

**Why**: explore\((u)\) starts and finishes within explore\((v)\)

**Back edges**: edge \((v,u)\) with

\[\text{pre}(u) < \text{pre}(v) < \text{post}(v) < \text{post}(u)\]

**Why**: explore\((v)\) starts and finishes within explore\((u)\)
cross edges: edge \((v,u)\) with 
\[ \text{pre}(u) < \text{post}(u) < \text{pre}(v) < \text{post}(v) \]

why: \(\text{explore}(u)\) finishes before \(\text{explore}(v)\) starts

How to decide if a directed graph \(G = (V,E)\) has a directed cycle:

Claim: \(G\) has a directed cycle \(\iff\) DFS-forest has a back edge.

Proof: \(\Leftarrow\) Assume \((v,u)\) is a back edge.

from page (83): in the solid tree, \(v\) is in the subtree of \(u\).

The tree edges from \(u\) to \(v\), plus edge \((v,u)\), form a directed cycle.
Assume $G$ has a directed cycle.

This cycle contains an edge $(v,u)$ such that $post(v) < post(u)$.

Why: Otherwise, $post(u) < post(v)$ for every edge $(v,u)$ on the cycle. Walk around the cycle, post-numbers get smaller and smaller and smaller...

$\therefore$ Escher's Ascending and Descending

Take an edge $(v,u)$ on the cycle with $post(v) < post(u)$.

From page 84: $(v,u)$ is not a tree edge and not a forward edge.

From top of page 85: $(v,u)$ is not a cross edge.

$\therefore (v,u)$ is back edge.
How to test if a directed graph is cyclic:
* Run DFS
* for each non-tree edge \((v, u)\), test if 
  \[ \text{pre}(u) < \text{pre}(v) < \text{post}(v) < \text{post}(u) \]
  if "yes" for at least one non-tree edge: cyclic,
  if "no" for all non-tree edges: acyclic.

**Running time:** \(O(|V| + |E|)\).

Assume \(G\) is acyclic. How to do topological sorting:
* Run DFS.
* Run Bucket-Sort to sort the vertices by post-number. \([\text{every post-number } \in \{2, 3, 4, \ldots, 2|V|\}]\]
* obtain the topological sorting from the reverse sorted order.

**Running time:** \(O(|V| + |E|)\).
Run DFS:

```
A ----> C ----> E
|         |         |
V         V         V
B ----> D ----> F
```

sort by post number: E, F, C, A, D, B

topological sort: reverse order:

B, D, A, C, F, E

1 2 3 4 5 6

all edges go from left to right
Correctness of algorithm at bottom of page 87:

Let \((v,u)\) be an edge of \(G\).

To show: in topological sorting,
\[\text{number of } v < \text{number of } u\]

Same as: \(\text{post}(v) > \text{post}(u)\). \(-\text{To show.}\)

Since \(G\) is acyclic: \((v,u)\) is not a back edge (from claim on page 85)

\[\therefore (v,u) \text{ is a tree edge, or a forward edge, or a cross edge.}\]

\[\therefore \text{ from page 84 and top of page 85 }:\]
\[\text{post}(v) > \text{post}(u)\].