Edge Classification (DFS and BFS)

- **Tree edge** if \( \text{pred}(v) = u \) (or \( \text{pred}(u) = v \) in undirected)
- **Back edge** if \( u \) is a descendant of \( v \) (or \( v \) is a descendant of \( u \) in undirected)
- **Forward edge** if \( u \) is an ancestor of \( v \) (directed only)
  - (never in BFS)
  - (finds shortest paths)
- **Cross edge** otherwise
  - (never in DFS on undirected)

For graph \( G \), \( G \) is acyclic if and only if no back edges in DFS(\( G \))

\[ \Rightarrow \] Suppose \( G \) is acyclic. Suppose \( \exists \) back edge \( u \rightarrow v \) [hope for contradiction]

- Then \( \exists \) tree edges \( v \rightarrow u \). dot of back edge
- So \( v \rightarrow u \rightarrow v \), which is a cycle \( \Rightarrow \)
- So no back edges,

\[ \Leftarrow \] Suppose no back edges in DFS of \( G \). [goal: \( G \) is acyclic]

- Suppose \( G \) has a cycle \( v_1, \ldots, v_k, v_1 \) [hope for a contradiction]
- Assume WLOG \( v_1 \) has lowest discovery time. (So \( v_1 \) discovered first)
- So at \( d[v_1] \), \( \text{COLOR}(v_1) = \cdots = \text{COLOR}(v_k) = \text{WHITE} \)
- \( v_k \) is a descendant of \( v_1 \) by While Path Thm
- \( (v_k, v_1) \) is a back edge \( \Rightarrow \)
- \( G \) is acyclic.