"What about the efficiency of this algorithm [DFS and BFS]? Our assumption is that the work done as we visit each node is the most complex part of the process. So, the work done to check to see if an adjacent node has been visited and the work to traverse the edges is not significant in this case. So the order of the algorithm is the number of times a node is visited. Because we have said these algorithms visit each node exactly one time, for a graph with \( N \) nodes, the visit process will be done \( N \) times. These traversals are, therefore, of order \( O(N) \). ... At worst, we will examine all \( |E| \) edges in the process of doing the traversal."

**DFS application #1**

\[
\text{running time } = 100 \cdot |V| + 1 \cdot |E| \\
\text{for dense graph } \quad 100 \cdot |V| + \frac{1}{2} |V|^2 \\
\text{for large enough } |V| \text{ this dominates}
\]
Nested Internal Thm

for any \( u, v \) \((u \neq v)\), either
1) \( d(u) \leq d(v) \leq f(v) \leq f(u) \) and \( v \) is a descendant of \( u \)
2) \( d(v) \leq d(u) \leq f(u) \leq f(v) \) and \( u \) is a descendant of \( v \)
3) \( d(u) \leq f(v) \leq d(v) \leq f(u) \) and no ancestor/descendant relationship

Proof: Assume \( u \neq v \) and WLOG \( d(u) \leq d(v) \)

Two cases: 1) \( f(u) \leq d(v) \)

\[ d(u) \leq f(u) \leq d(v) \leq f(v) \]

\( u \) is BLACK at time \( d(v) \)

so backtracked from \( u \) and in diff part of first case of NIT

2) \( d(v) \leq f(u) \)

\( u \) is LARGE at time \( d(v) \)

DFS-Visit \((u)\) is in charge of tree \((v)\)

DFS-Visit \((u)\) is an ancestor of DFS-Visit \((v)\)

\( u \) is an ancestor of \( v \)

\[ f(u) \leq f(v) \leq f(u) \]

case 1 of NIT

White Path Thm: \( d(u) \leq d(v) \leq f(v) \leq f(u) \)

if and only if \( v \) is a descendant of \( u \)

by case 1 of NIT

there exists a path of WHITE with \( u \rightarrow v \) at time \( d(u) \)

\( \Rightarrow (\triangleright) \): Suppose \( d(u) \leq d(v) \leq f(v) \leq f(u) \). Then \( v \) is not a descendant of \( u \)

when path is tree edges \( u \rightarrow v \): all nodes on path are descendants of \( u \), so by NIT (case 1), all have \( d(x) \geq d(u) \), so all are WHITE at time \( d(u) \)

\( \Leftarrow (\triangleright) \): Suppose \( V_1, \ldots, V_k \) is a white path at time \( d(u) \)

Then \( d(v) \leq d(u) \) \( \triangleright \) \( V \) is WHITE at time \( d(u) \)

Suppose \( v \) is not a descendant of \( u \), looking for contradiction \( \Rightarrow \)

Then \( d(u) \leq f(u) \leq d(v) \leq f(v) \) case 3 of NIT

Find 1st \( v_i \) in path s.t. \( d(u) \neq d(v_i) \leq f(v_i) \leq f(u) \)

Note \( 2 \leq i \leq k \)

So \( d(u) \leq d(v_{i-1}) \leq f(v_{i-1}) \leq f(u) \leq d(v_i) \leq f(v_i) \)

\( v_{i-1} \) must be in case 1 of NIT between times \( d(v_{i-1}), f(v_{i-1}) \), \( \text{color}(v_i) = \text{WHITE} \)

so edge \((v_{i-1}, v_i)\) is checked, \( \text{color}(v_i) = \text{WHITE} \)

\( v_i \) is a child of \( v_{i-1} \) which is a descendant of \( u \)

\( v_i \) is a descendant of \( u \)

\( d(u) \leq d(v_i) \leq f(v_i) \leq f(u) \) \( \Rightarrow \)

\( v \) is a descendant of \( u \)
\[ d(v) < d(u) < f(v) < f(u) \] by case 1.