As in the text, let $v.d$ and $v.f$ denote the discovery and finish times, respectively, of vertex $v$ in the depth-first search algorithm.

1. Suppose $v$ is a descendant of $u$ in a $G_\pi$-tree created by the depth-first-search algorithm. How are the discovery and finish times of $u$ and $v$ related?

2. Recall that the depth-first search algorithm colors vertices as it progresses. Consider the point when vertex $v$ is scanned on the adjacency list of vertex $u$. What color is $v$ if $(u, v)$ would form a back edge in the $G_\pi$ tree containing $u$ and $v$?

3. In a depth-first search on a acyclic graph, the vertices arranged in decreasing $v.f$ order produce a topological sort. Why?

4. The Kruskal and Prim algorithm both generate minimal spanning trees by adding safe edges to an initially empty set. However, they choose their safe edges in different ways. What is the difference?

5. Suppose $G_n = (V_n, E_n)$ is a sequence of graphs of increasing size. That is, $V_{n+1} + E_{n+1} > V_n + E_n$. As the graphs grow, they maintain the relationship $E = \Theta(V^{1/2})$. Let $f(n)$ be the size of the adjacency list representation for $G_n$. Let $g(n)$ be the size of the adjacency matrix representation for $G_n$. Which of $o(g), O(g), \Theta(g), \Omega(g), \omega(g)$ is the best classification of $f$ and why?
Solutions.

1. Suppose \( v \) is a descendant of \( u \) in a \( G_\pi \)-tree created by the depth-first-search algorithm. How are the discovery and finish times of \( u \) and \( v \) related?

Parenthesis Theorem: \( u.d < v.d < v.f < u.f \).

2. Recall that the depth-first search algorithm colors vertices as it progresses. Consider the point when vertex \( v \) is scanned on the adjacency list of vertex \( u \). What color is \( v \) if \( (u, v) \) would form a back edge in the \( G_\pi \) tree containing \( u \) and \( v \)?

Back edge \( (u, v) \) places \( v \) higher in the same \( G_\pi \) tree than \( u \). Therefore, the adjacency list of \( v \) is still being scanned in a suspended stack frame underneath the active frame that is scanning the adjacency list of \( u \). Consequently, \( v \) is gray.

3. In a depth-first search on a acyclic graph, the vertices arranged in decreasing \( v.f \) order produce a topological sort. Why?

We need to show that for any edge \( (u, v) \), we must have \( v.f < u.f \). That is, all edges point forward in the decreasing finish-time order. We make the following points.

(a) An acyclic graph can contain no back edges because a back edge runs from descendant to an ancestor in a \( G_\pi \) tree, which would constitute a cycle when followed by the path of tree edges connecting the ancestor to the descendant.

(b) When \( v \) is scanned on the adjacency list of \( u \), \( v \) cannot be gray, since gray denotes a back edge.

(c) If \( v \) is white, then \( v \) is newly discovered at this time, while \( u \) has not yet finished. Because discovery-finish interval must be either nested or disjoint, we must have \( u.d < v.d < v.f < u.f \). That is, \( v.f < u.f \) as desired.

(d) If \( v \) is black, then \( v.f \) has been assigned, but since \( u \) has not yet finished, its later assignment will place \( v.f < u.f \).

4. The Kruskal and Prim algorithm both generate minimal spanning trees by adding safe edges to an initially empty set. However, they choose their safe edges in different ways. What is the difference?

Kruskal chooses a light edge that connects two components (trees) in an evolving forest. Such an edge is safe because it is a light edge crossing the cut that isolates one of the components.

Prim chooses a light edge that connects an isolated vertex to a growing tree. Such an edge is safe because it is a light edge crossing the cut that separates the evolving tree from the isolated vertices that have not yet joined the tree.

5. Suppose \( G_n = (V_n, E_n) \) is a sequence of graphs of increasing size. That is, \( V_{n+1} + E_{n+1} > V_n + E_n \). As the graphs grow, they maintain the relationship \( E = \Theta(V^{1/2}) \). Let \( f(n) \) be the size of the adjacency list representation for \( G_n \). Let \( g(n) \) be the size of the adjacency matrix representation for \( G_n \). Which of \( o(g), O(g), \Theta(g), \Omega(g), \omega(g) \) is the best classification of \( f \) and why?

\[
\begin{align*}
f(n) & \leq K_1(V + E) \\
g(n) & \geq K_2V^2 \\
f(n) & \leq K_1(V + E) \\
g(n) & \leq \frac{K_1V + K_1K_3V^{1/2}}{K_2V^2} \\
E & \leq K_3V^{1/2} \\
f(n) & \leq \frac{K_1V + K_1K_3V^{1/2}}{K_2V^2} \to 0 \\
f & = o(g).
\end{align*}
\]