BFS: Assume we are given the undirected graph below, and that lines 1-9 of the BFS(G,s) algorithm have been executed. In particular, all the vertices are colored white with the exception of the initial vertex s = v1 which has been colored gray. Also assume that the vertex v1 has been pushed onto the queue Q.

\[
\text{Q = v1}
\]

\textbf{u = v1: Dequeue v1}

Walk through G.Adj[v1] list = v2 and v3

\[
\begin{align*}
\text{v=v2: v2.color = GREY} \\
\text{v2.d = 1} \\
\text{v2.π = v1} \\
\text{Q = v2}
\end{align*}
\]

\[
\begin{align*}
\text{v=v3: v3.color = GREY} \\
\text{v3.d = 1} \\
\text{v3.π = v1} \\
\text{v1.color = BLACK} \\
\text{Q = v3, v2}
\end{align*}
\]

\textbf{u = v2: Dequeue v2}

Walk through G.Adj[v2] list = v1 and v4

\[
\begin{align*}
\text{v=v1: v1.color != WHITE} \\
\text{v=v4: v4.color = GREY} \\
\text{v4.d = 2} \\
\text{v4.π = v2} \\
\text{v2.color = BLACK} \\
\text{Q = v4, v3}
\end{align*}
\]
\( u = v_3: \) Dequeue \( v_3 \)
Walk through \( G.\text{Adj}[v_3] \) list = \( v_1 \) and \( v_4 \)

\( v = v_1: \) \( v_1.\text{color} \neq \text{WHITE} \)
\( v = v_4: \) \( v_4.\text{color} \neq \text{WHITE} \)

\( v_3.\text{color} = \text{BLACK} \) \( Q = v_4 \)

\( u = v_4: \) Dequeue \( v_4 \)
Walk through \( G.\text{adj}[v_4] \) list = \( v_2, v_3, \) and \( v_5 \)

\( v = v_2: \) \( v_2.\text{color} \neq \text{WHITE} \)
\( v = v_3: \) \( v_3.\text{color} \neq \text{WHITE} \)
\( v = v_5: \) \( v_5.\text{color} = \text{GREY} \)
\( v_5.d = v_4.d + 1 = 3 \)
\( v_5.\pi = v_4 \)
\( Q = v_5 \)

\( v_4.\text{color} = \text{BLACK} \)

\( u = v_5: \) Dequeue \( v_5 \)
Walk through \( G.\text{adj}[v_5] \) list = \( v_4 \)

\( v = v_4: \) \( v_4.\text{color} \neq \text{WHITE} \)

\( v_5.\text{color} = \text{BLACK} \)

\( Q = 0. \) HALT

Note this process produces a tree with edges \( \{(\pi[v], v): v \neq s\} \)
DFS:

DFS(G): Stack: 12345

DFS-VISIT(G,v1):
  time = time + 1 = 1
  v1.d = 1
  v1.color = GRAY

G.adj[v1] list = v2, v3:
  v2.π = v1
  DFS-VISIT(G,v2):
    time = time + 1 = 2
    v2.d = 2
    v2.color = GRAY
    v4.π = v2
    DFS-VISIT(G,v4):
      time = time + 1 = 3
      v4.d = 3
      v4.color = GREY
      v1 = v3

DFS-VISIT(G,u)
  time = time + 1
  u.d = time
  u.color = GRAY
  for each v ∈ G.Adj[u] // discover u
    if v.color == WHITE // explore (u, v)
      DFS-VISIT(v)
  u.color = BLACK
  time = time + 1
  u.f = time // finish u
\(G.\text{adj}[v4]\) list = \(v2, v3, v5\): time = 4

DFS_VISIT(G,v2): \(!\text{WHITE}\)

\[V3.\pi = v4\]

DFS_VISIT(G,v3):

\[\text{time} = \text{time} + 1 = 4\]

\[v3.d = 4\]

\[v3.\text{color} = \text{GREY}\]

(1 2 4 3)

NOTE: At this point, no vertices adjacent to \(v3\) are white, so no more recursive calls within DFS_VISIT(G,v3) so execute lines 8-10 of DFS-VISIT

\[v3.\text{color} = \text{BLACK},\]

\[\text{time} = \text{time} + 1 = 5\]

\[v3.f = 5\]

(1 2 4 3 3)

\[V5.\pi = v4\]

DFS_VISIT(G,v5):

\[\text{time} = \text{time} + 1 = 6\]

\[v5.d = 6\]

\[v5.\text{color} = \text{GREY}\]

(1 2 4 3 3 5)

NOTE: As above, no vertices adjacent to \(v5\) are white, so no more recursive calls within DFS_VISIT(G,v5) so execute lines 8-10:

\[v5.\text{color} = \text{BLACK}\]

\[\text{Time} = \text{time} + 1 = 7\]

\[v5.f = 7\]

(1 2 4 3 3 5 5)
Note also at this point that we have iterated through the adjacency list for v4, so execute lines 8 – 10:

\[
\begin{align*}
V4.color &= \text{BLACK} \\
\text{time} &= \text{time} + 1 = 8 \\
v4.f &= 8 \\
(1 \ 2 \ 4 \ (3 \ 3) \ (5 \ 5) \ 4)
\end{align*}
\]

Similarly, we have finished iterating through the adjacency list for v2, so can execute lines 8-10:

\[
\begin{align*}
V2.color &= \text{BLACK} \\
\text{time} &= \text{time} + 1 = 9 \\
v2.f &= 9 \\
(1 \ 2 \ 4 \ (3 \ 3) \ (5 \ 5) \ 4) \ 2)
\end{align*}
\]

And finally, we are finished with the adjacency list for v1, so can execute lines 8-10:

\[
\begin{align*}
V1.color &= \text{BLACK} \\
\text{Time} &= \text{time} + 1 = 10 \\
v1.f &= 10 \\
(1 \ (2 \ 4 \ (3 \ 3) \ (5 \ 5) \ 4) \ 2) \ 1)
\end{align*}
\]

Since no vertices are colored white, DFS(G) makes no more calls to DFS-VISIT.

Halt.