Notes on DFS

We use $v \Rightarrow w$ to mean there is a directed path from $v$ to $w$. This includes the case $w \in \text{Adj}(v)$ but the path can be of any length.

Theorem (I like to call this the White Path Theorem): Let $v = z_0, \ldots, z_r = w$ be a path from $v$ to $w$. (There may be many.) Let $v$ be discovered at time $t$. Assume that at time $t$ the points $z_1, \ldots, z_r = w$ are all white. (We will call this a white path from $v$ to $w$.) Then (this is the result we claim) $\text{DFS} - \text{VISIT}(v)$ will reach $w = z_r$.

One wrong proof is to simply say that $\text{DFS} - \text{VISIT}(z_0)$ goes to $z_1, z_2, \ldots$ until reaching $z_r = w$. We have to be careful here. While $z_1 \in \text{Adj}(z_0)$ it might be late in the adjacency list and it might already have been discovered when it is come to. We gave an informal argument in class on April 2. The formal argument is by induction.

$r = 1$. That is $w \in \text{Adj}(v)$. Now during $\text{DFS} - \text{VISIT}(v)$ we will go through the adjacency list of $v$ and at some point reach $w$. It may be that $w$ was reached earlier – perhaps there is a path $v_rw$ and $r$ came earlier in $\text{Adj}(v)$ – but it will be reached during $\text{DFS} - \text{VISIT}(v)$.

Induction on $r$. Assume for all shorter path lengths. We know from above that $z_1$ will be reached. (But, playing devil’s advocate, maybe $z_2$ was reached before $z_1$. So when $z_1$ started through its adjacency list it did nothing with $z_2$ because it wasn’t white.) Here is the key idea: Let $z_I$ be the first (in time) of $z_1, \ldots, z_r$ to be reached. If $I = r$, $w$ is reached, we are done. Else consider the situation when $z_I$ is reached. There is a path $z_I, z_{I+1}, \ldots, z_r = w$ from $z_I$ to $w$. When $z_I$ is reached the points $z_{I+1}, \ldots, z_r = w$ are still white. (Critically, $z_I$ was the first one reached!) Now $\text{DFS} - \text{VISIT}(z_I)$ is called. By induction on the path length (as the path from $z_I$ to $w$ is shorter than $r$) $w = z_r$ will be reached during $\text{DFS} - \text{VISIT}(z_I)$. But $\text{DFS} - \text{VISIT}(v)$ has $\text{DFS} - \text{VISIT}(z_I)$ inside it so $w$ is reached during the (longer) $\text{DFS} - \text{VISIT}(v)$. This concludes the proof.

Now we can fully describe how the colors of the vertices change from the start (time $d(v)$) of $\text{DFS} - \text{VISIT}(v)$ to the end (time $f(v)$) of $\text{DFS} - \text{VISIT}(v)$. If $z$ was Gray or Black then it is unaffected, its color remains the same. If $z$ was White there are two cases. If there was a white path from $v$ to $z$ then, by the above result, $z$ was reached. Therefore $\text{DFS} - \text{VISIT}(z)$ is called inside of $\text{DFS} - \text{VISIT}(v)$. Therefore $\text{DFS} - \text{VISIT}(z)$ is finished before $\text{DFS} - \text{VISIT}(v)$. Therefore $z$ will be black at time $f(v)$. Else: There is no white path from $v$ to $z$. Then $z$ will not be reached during $\text{DFS} - \text{VISIT}(v)$. Therefore $z$ will remain white. (Notice: that no $z$ that
had been white ends up gray!)

This is particularly nice at the start of DFS. Suppose \( v \) is the first vertex of \( V \) (remember, \( V \) is given by a linked list). When we start \( DFS - VISIT(v) \) all \( z \neq v \) are white. So when we finish \( DFS - VISIT(v) \) there are only two cases. Either \( v \rightarrow z \) in which case \( z \) will be black. Or not – in which case \( z \) will remain white. So the black vertices are precisely those \( z \) for which \( v \rightarrow z \). This has a useful corollary: If \( z \) is black and \( z \rightarrow z' \) then \( z' \) will be black. Why? Because \( v \rightarrow z \) and \( z \rightarrow z' \) and hence \( v \rightarrow z' \) and hence \( z' \) will be black.

Now let’s look at DFS as a whole, splitting the time cleverly. We take the first vertex, call it \( v_1 \) and apply \( DFS - VISIT(v_1) \). Now all vertices are white or black – \( z \) black if \( v_1 \rightarrow z \). Now DFS skips over the nonwhite vertices until it reaches a white vertex, call it \( v_2 \), and applies \( DFS - VISIT(v_2) \). Now \( z \) turns black iff there is a white path from \( v_2 \) to \( z \).

Claim: If \( v_2 \rightarrow z \) then \( z \) will be black by the end of \( DFS - VISIT(v_2) \).

Proof: If the path from \( v_2 \) to \( z \) is a white path then this is the white path theorem. But suppose the path from \( v_2 \) to \( z \) included some black point \( y \). (To clarify: that \( y \) was black after \( DFS - VISIT(v_1) \).) Then \( y \rightarrow z \) so, from the corollary in the above paragraph, \( z \) was already black at the end of \( DFS - VISIT(v_2) \). Either \( z \) was black or became black, either way it is black.

That is, after \( DFS - VISIT(v_1) \) and \( DFS - VISIT(v_2) \) all \( z \) that have either \( v_1 \rightarrow z \) or \( v_2 \rightarrow z \) are black and all other \( z \) are white. This continues. (The Math/CS majors can write the induction argument!) After \( DFS - VISIT(v_i), 1 \leq i \leq r \), all \( z \) are white or black. \( z \) is black if some \( v_i \rightarrow z \), else it is white.

We now apply these ideas to the algorithm for finding a strongly connected component. We split the (arbitrary) directed graph \( G \) into strongly connected components \( C_1, C_2, \ldots, C_s \). We write \( C_i \rightarrow C_j \) if there is some \( v_i \in C_i, v_j \in C_j \) with \( v_j \in \text{Adj}[v_i] \). (Effectively, we are “squishing” the \( C_i \) into supernodes and creating a directed graph on them.) As shown in class there can be no cycle on these supernodes. Call \( C_i \) a top component (there may be many of them) if there is no \( C_j \) with \( C_j \rightarrow C_i \).

Now apply DFS to \( G \) and let \( v \) be the vertex that finishes last.

Claim: \( v \in C \) where \( C \) is a top component.

Proof: Suppose there was a \( C' \rightarrow C \). From our above description \( v = v_r \) where DFS has consisted of \( DFS - VISIT[v_i], 1 \leq i \leq r \). Could any \( v' \in C' \) have been visited during one of the \( DFS - VISIT[v_i], 1 \leq i \leq r - 1 \)? No! Because if \( v' \) turns black then all \( w \) with \( v' \rightarrow w \) turn black and that would include \( v \). So when we started \( DFS - VISIT[v] \) the nodes \( v' \in C' \)
were still white. But as \( C' \to C \) there is no path \( v \Rightarrow v' \) and therefore \( DFS - VISIT[v] \) leaves \( C' \) white. So DFS is not over, \( v \) is not the last vertex to finish.

Example: Strongly connected components \( A, B, C, D \) with \( A \to B, A \to C, B \to D \) and \( C \to D \). Say \( v_1 \in B \) Then \( DFS - VISIT[v_1] \) turns \( B, D \) black. The nonblack vertices now consist of \( A, C \) with \( A \to C \). Now say \( v_2 \in C \). Then \( DFS - VISIT[v_2] \) turns \( C \) black. (Note that \( D \) is already black.). Now the nonblack vertices consist solely of \( A \). DFS hits \( DFS - VISIT[v_3] \) with \( v_3 \in A \). All of \( A \) turns black – DFS is effectively over (it doesn’t find any more white vertices) and \( v_3 \) was the last to finish. (Note that DFS might proceed in a different order. It may be that \( v_1 \in A \). Then \( DFS - VISIT[v_1] \) turns everything black and \( v_1 \) finishes last. But no matter what the order for this example the vertex finishing last has to be in \( A \).)