CS 344: Computation of Strongly Connected Components (SCC)

Let $G = (V, E)$ be a directed graph. Strongly connected components are maximal subgraphs having the property that any pair of vertices in them can be reached from each other via directed paths. To obtain SCC's we first compute a DFS traversal. For each $u \in V$, we define its forefather, denoted by $\phi(u)$, to be the vertex that satisfies two criteria: (1) it can be reached from u via any directed path (independent of DFS trees), (2) its finishing time (dependent on the DFS) is maximum. We can show that $\phi(u)$ is an ancestor of u. In other words not only u can reach $\phi(u)$, but the converse is also true. Thus u and $\phi(u)$ are in the same SCC (the ancestor-descendant relationship is defined with respect to the given DFS and it implies there is a path from ancestor to descendant). Also, we can show that u and v are in the same SCC if and only if they have the same forefather. This means that all we have to do is to recognize the forefathers and those that can reach them. If we pick the vertex r having the largest finishing time value $f(r)$, it must be a forefather. To find all the vertices in G which can reach r, i.e. in the same SCC, we first reverse direction in G to get $G^T = (V, E^T)$, where $E^T = \{(u, v) : (v, u) \in E\}$. Then, we do a DFS on G^T starting at r. This gives SCC of r. Next we pick a vertex unvisited by DFS of G^T with highest finishing time and repeat. The SCC algorithm is:

1. Call DFS(G) to compute $f(u)$ for all $u \in V$.
2. Compute G^T, and call DFS(G^T), but in main loop of DFS, consider vertices in order of decreasing $f(u)$.
3. Output vertices of each tree in DFS forests of step two. These are the SCC's.