CIS 575. Introduction to Algorithm Analysis
Material for April 17, 2024

Algorithms Based on Depth-First Search

The topic of this note is part of what is covered in *Cormen*'s Section 20.3.

1 Generic Depth-First Search

We shall *augment* our algorithm for depth-first search so that it can take various kinds of action at various points: when a node has just been discovered; when a tree edge is about to be explored; when a tree edge has been explored; when a non-tree edge is encountered; when a node is finished. We get the algorithm

```
DFS(u)  // u is white
    color[u] ← gray; d[u] ← current_time
    PreNode action
    foreach w where E contains an edge from u to w
      if color[w] = white
        PreEdge action
        T ← T ∪ {u → w}
        DFS(w)
        PostEdge action
      else
        OtherEdge action
    color[u] ← black; f[u] ← current_time
    PostNode action
```

By suitable instantiation of the 5 kinds of action, numerous useful graph algorithms may be constructed. In the subsequent notes, we shall present two such instantiations: **Topological sort** for **directed** graphs (the topic of *Cormen*'s Section 20.4), and **Articulation points** for **un**directed graphs (as mentioned in *Cormen*'s Problem 20-2).

Running Time. Assuming that the various actions each run in *constant* time, we see that the total time spent within the **foreach** loop, not counting the recursive calls, is in $\Theta(|E|)$, and that the total time spent elsewhere is in $\Theta(|V|)$. Thus the generalized DFS algorithm runs in time $\mathbf{\Theta}(|\mathbf{V}| + |\mathbf{E}|)$ which for a **connected** undirected graph, or a **strongly connected** directed graph, amounts to $\mathbf{\Theta}(|\mathbf{E}|)$.