Lecture 17, Nov 13, 2023

Depth First Search

- In DFS, the last discovered node is the first explored
- Like BFS, we will track the color: white/grey/black, and parent $p[v]$
- DFS also tracks $d[v]$, the "discovery time" of v, and $f[v]$, the "exploration complete time" of v
  - To keep track of time we use a counter which is incremented by 1 every time we discover a node and finish exploring a node
- DFS Algorithm:
  - Initialization: set every color to white, $d[v], f[v]$ to infinite, $p[v]$ to nil, time to 0
  - For every node v, if v is white, call the recursive DFS-Explore(G, v) on the node
- The DFS-Explore(G, u) procedure explores all nodes reachable from u:
  - Color u grey, increment $time = time + 1$, set $d[v] = time$
  - For each node v connected to u, if v is white, set $p[v] = u$ and call DFS-Explore(G, v)
  - Once all connected nodes are explored, color u black, increment $time = time + 1$, set $f[v] = time$
- Note that $f[v]$ for the final node is exactly $2V$, since every node contributes 2 to the time
- The time complexity is also $O(V + E)$ like BFS
- The tree formed by all edges in $p[v]$ (i.e. all discovery edges) is the DFS forest, similar to the BFS forest
  - Edges in the original graph that are included in the DFS forest are tree edges
- A non-tree edge can be one of 3 types:
  - Forward edge: going from an ancestor to a descendant in the DFS forest
  - Back edge: going from a descendant to an ancestor in the DFS forest
  - Cross edge: neither forward nor backward edges (i.e. between two different subtrees)
- Note some properties of $d[v]$ and $f[v]$:
  - If u is an ancestor of v, $d[u] < d[v] < f[v] < f[u]$
  - For any 2 nodes u and v, we can never have $d[u] < d[v] < f[u] < f[v]$
  - If there is an edge from u to v, we will always have $d[v] < f[u]$
  - For any tree or forward edge (u, v), $d[u] < d[v] < f[v] < f[u]$ since u is an ancestor of v
  - For any back edge (u, v), $d[v] < d[u] < f[u] < f[v]$ since v is an ancestor of u
  - For any cross edge (u, v), $d[v] < f[v] < d[u] < f[u]$
    - Since u and v have no relation, and we cannot have $d[u] < d[v] < f[u] < f[v]$, we must have either $d[v] < f[v] < d[u] < f[u]$ or $d[u] < f[u] < d[v] < f[v]$, but the latter is impossible since there is an edge (u, v)
- Note DFS is not unique and it depends on the node we starts from