This and other notes are available from `http://www.cs.berkeley.edu/~yozo/cs170.fa05/`

Depth First Search

- Edge Classification
  In DFS, any edge $e = (u, v)$ is classified as one of following:
  - Tree Edge
    Vertex v was first seen through this edge.
    $pre[u] < pre[v] < post[v] < post[u]$.
  - Back Edge
    Discovered at u going back up to the parent vertex v which in process of being explored.
    $pre[v] < pre[u] < post[u] < post[v]$.
  - Forward Edge
    Discovered at u going down to a descendent that exploration came back from.
    $pre[u] < pre[v] < post[v] < post[u]$.
  - Cross Edge
    Connects two branches of exploration.
    $pre[v] < post[v] < pre[u] < post[u]$.
- Parenthesis Property In DFS, if u, v are any vertices, then the intervals $(pre[u], post[u])$ and $(pre[v], post[v])$ are either disjoint, or one entirely contains the other.
- Applications
  - Topological Sort
    sort vertices in decreasing order of post-visit number.
  - Strongly Connected Components
    run DFS on G^R, run DFS on G considering the vertices in decreasing order of post-visit number.
  - Cycle Detection
    detect back edge

Questions

1. In a clique of size 7, how may edges are there? How may triangles? Generalize.
2. Which of four types of edge can occur in a DFS of an *undirected* graph?
3. Is a topological sort of a graph G unique? Why or why not?
4. Prove (formally) that topological sort algorithm given in class works.
5. Give a convincing argument why the parenthesis property might hold.
6. Explain how DFS and BFS are quite similar; one uses queue instead of stack.