Tutorial 9, Nov 17, 2023

Topological Sorting

- Suppose that in a directed graph $G = (V, E)$ where nodes represent tasks, and each edge (u, v) means that task u must be done before task v; how should we order the tasks such that the order constraints are satisfied?
  - We can note that this is impossible if there exists a cycle in a graph
- This is the problem of *topological sorting*: given a directed acyclic graph (DAG) G, find an ordering of vertices in V such that for every directed edge (u, v) in G, u appears before v in the ordering
- Claim: we can obtain a topological sort of G by performing a DFS on G; as each vertex is finished, we insert it into the start of a linked list; at the end of the DFS, the linked list will contain a topological sorting of the nodes
  - Note that since we insert at the front of the list, we essentially return the nodes in reverse order as they are finished
  - This has complexity $\Theta(m + n)$
- Proof:
  - The procedure gives back the nodes of G in order of decreasing finishing times
  - Therefore it suffices to show that for every edge (u, v) of G, $f[u] > f[v]$ in the DFS (this would mean that u comes before v in the topological sort)
  - First way: Consider the color of v just before the edge (u, v) is explored:
    * If v was grey, then v must be an ancestor of u, since being grey means it is in the current path; so (u, v) is a back edge, but this would mean G has a cycle, which cannot happen
    * If v is white, then v will become a descendant of u and (u, v) is a tree edge; therefore we have $d[u] < d[v] < f[v] < f[u]$
    * If v is black, then v will have already been explored, so $f[v] < f[u]$
  - Second way: Let (u, v) be an edge; we know it cannot be a back edge since G has no cycles; consider all cases:
    * If (u, v) is a tree edge or forward edge, then v is a descendant of u, so $d[u] < d[v] < f[v] < f[u]$
    * If (u, v) is a cross edge, we claim $d[v] < f[v] < d[u] < f[u]$, i.e. v was completely done before u was discovered
      - By definition of cross edge, u and v do not have a descendant relationship
      - We cannot have $d[u] < d[v] < f[v] < f[u]$ or $d[v] < d[u] < f[u] < f[v]$, so the intervals are either entirely disjoint or crossed
      - We also cannot have $d[u] < d[v] < f[u] < f[v]$ or $d[v] < d[u] < f[v] < f[u]$, because this cannot happen in a DFS
      - Therefore our only options are $d[u] < f[u] < d[v] < f[v]$ or $d[v] < f[v] < d[u] < f[u]$
      - However, we can't have $f[u] < d[v]$, because of the edge (u, v) we must have that v was discovered before u finished