CS301 LinXiLi   ~~P4~~ P13

22.5

Strongly-connected-components-modified (G)

1 call DFS(G) to compute finishing times u.f for each vertex u

2 compute G^T

3 call DFS(G^T), in main loop of DFS, consider the vertices in order of decreasing u.f.

Output the vertices of each tree in the depth-first forest formed in line 3 as separate strongly connected components.

for ~~all~~ each vertex in V:
  vertex = 1 to k

for each vertex in V.
  ~~If vertex.adj[i]~~ have no edge with vertex.
    ~~add edge~~.

  If Adj[vertex] ~~vertex.adj[i]~~ have no edges,
    add edge to Adj[vertex] and vertex

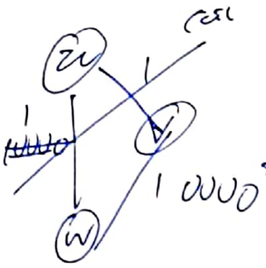
23.1-11

~~Given a~~ E = newly-added-decreased_weight edge
  V = E.V.
 for V.adj:
  If No edge in V.adj.
    add edge.
 And we get a graph here (minimum spanning tree with a new edge and a cycle.)
   MST(Graph).

扫描全能王 创建

23.2-8

suppose there are only three vertexes. And the edge's weight differs very big.



And we do the cut in the picture

we can easily do the MST between v, w is 10000, but when we connect u with this MST. the total weight is ~~10000~~ 10001. But the actually MST simply can be 2.