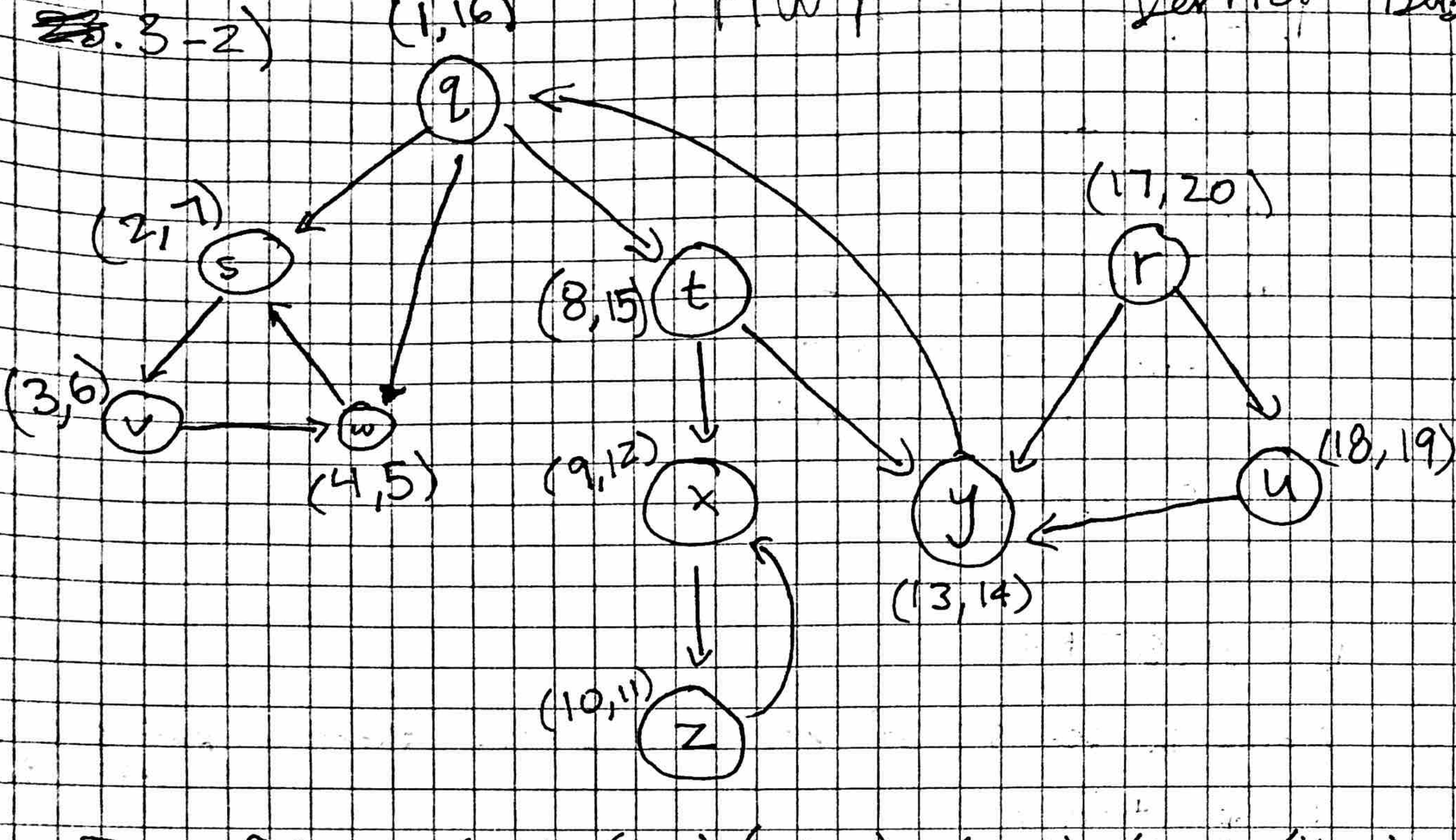
HW7 Derrick DeBose

22.3-2)



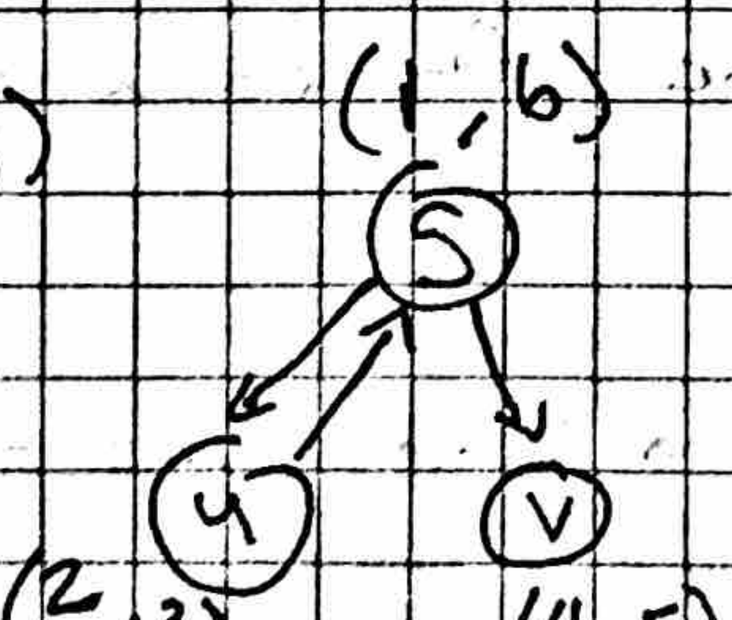
Tree Edges: (q,s), (s,v), (v, w), (q,t), (t,x), (x,z), (t,y), (r,u)

Back Edges: (w,s), (z,x), (y,q)

Forward edges: (q,w)

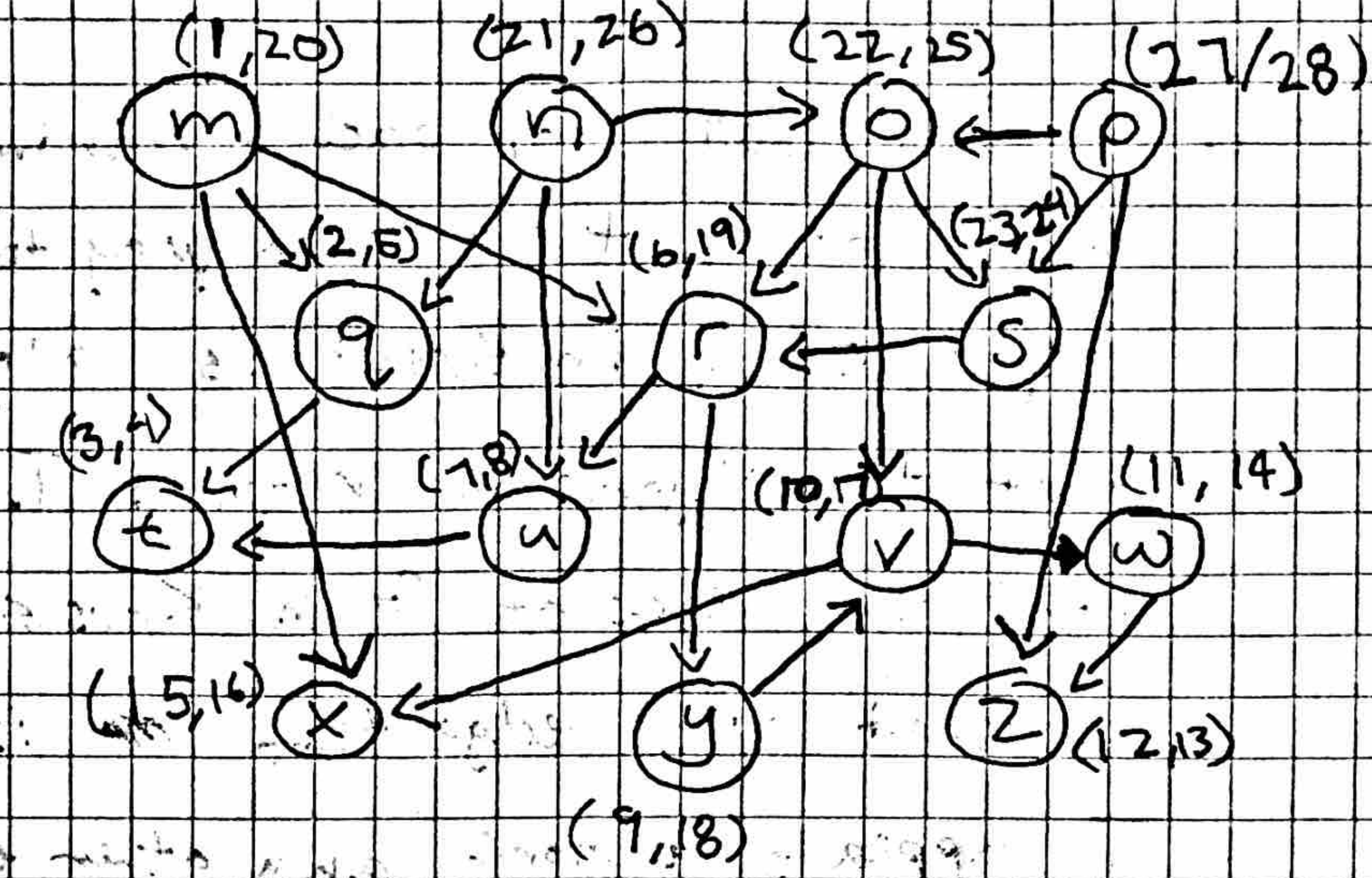
Cross edges: (r,y), (u,y)

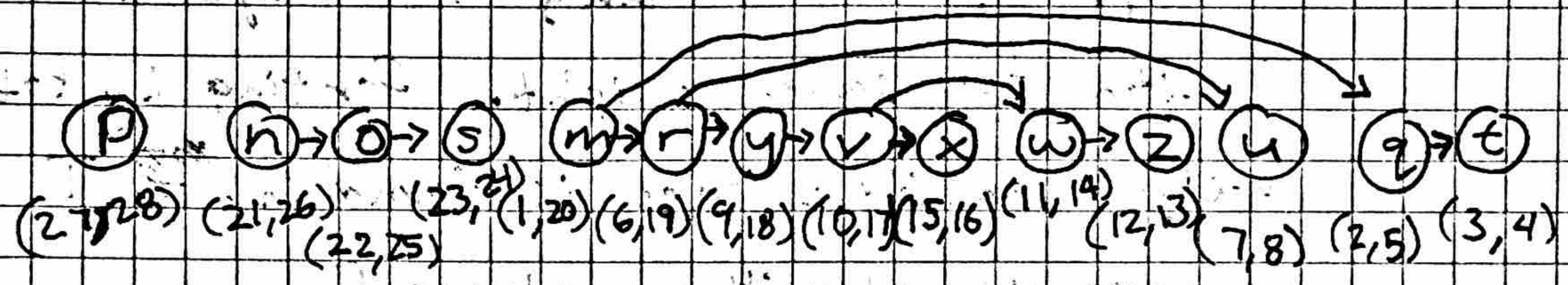
22.3-8)



v is <u>not</u> a decendent of u in a depth-first forest because $u.d < u.f < v.d < v.f$.

22.4-1)





23.1-1) Using the generic MST algorithm if we choose a cut such that u is on one side of the cut and v on the other. Then (u,v) is a light edge that's crossing a cut and its weight is the minimum of any edge crossing the cut. So, it's safe to add (u,v). So (u,v) does belong to some minimum spanning tree.

23.1-3) if (u,v) is an edge contained in some minimum spanning tree, then it is a light edge crossing some cut of the graph. if (u,v) edge is removed and draw a cut cross (u,v). Since u and v are on different sides of the min spanning trees then the (u,v) edge is a light edge

23.2-1) Show that for each minimum spanning tree T of G, there is a way to sort the edges of G in Kruskal's algorithm so that the algorithm returns T.

Sort the edges of G in Kruskal's algorithm so that every edge that is found in T appears before any other edge not in T with the same weight. Sorting by both weight and if the edge is found in T ensures the algorithm returns T.