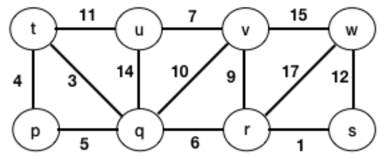
Problem 1. For the two parts below, use the following graph:



(a) Run Prim's algorithm starting from vertex p. Show the edges in the order in which they are added to form a minimum spanning tree.

(b) Run Kruskal's algorithm to find the minimum spanning tree. Show the edges in the order in which they are added to the minimum spanning tree.

Problem 2. Let $G = (V, E)$ be a directed graph.

1. Assuming that G is represented by a 2-dimensional adjacency matrix $A[1, \ldots, n,\ 1, \ldots, n]$, give a $\theta(n^2)$-time algorithm to compute the adjacency list representation of G, with $A[i, j]$ representing an edge between i and j vertices. (Represent the addition of an element(vertex), v, to an adjacency list, l, using pseudo-code, $l \leftarrow l \cup \{v\}$.)
2. Assuming that G is represented by an adjacency list $Adj[1, \ldots, n]$, give a $\theta(n^2)$-time algorithm to compute the 2-dimensional adjacency matrix representation of G.

Problem 3. Give a linear time, depth-first-search algorithm to find the size of the largest connected component in a graph, where size is measured by the number of edges in the component. (This should be a small modification to the DFS algorithm covered in class.). You may just print the final size.