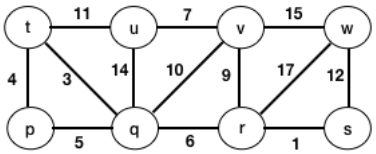
Problem 1. For all three parts below, use the following graph:



(a) Apply Dijkstra's algorithm using vertex p as the source vertex. Write the vertices in a table, in the order in which they are marked. In another table report all distances at each step of the algorithm. The first row of the order of the marked vertices table and the first row of the distances table are shown below as well.

Marked Vertices

step	vertex
1	p

Distances

step	dist[p]	dist[q]	dist[r]	dist[s]	dist[t]	dist[u]	dist[v]	dist[w]
1	0	5	∞	∞	4	∞	∞	∞

(b) 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.

(c) 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. Design an efficient algorithm to find a universal sink, i.e., a vertex with in-degree $n-1$ and out-degree 0 in a directed graph. The graph is given to you in an adjacency matrix representation and the weight of each edge is 1. How many exact cells would you need to check in the adjacency matrix?