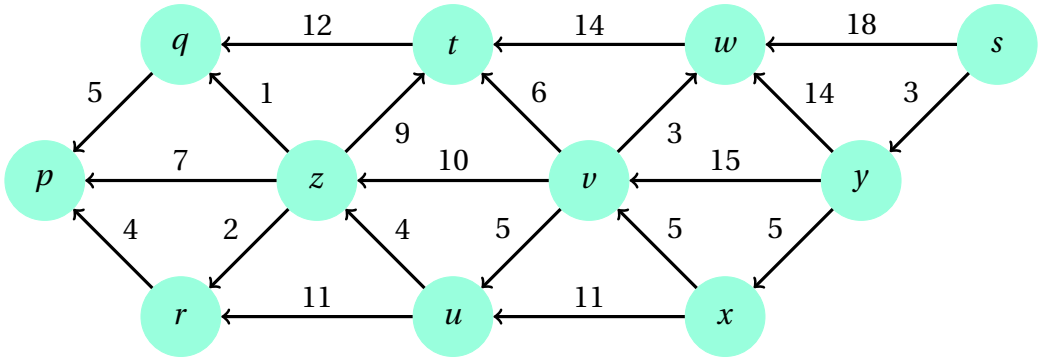
Week 6 Preparation

(Solutions)

Useful advice: The following solutions pertain to the preparation problems. You are strongly advised to attempt the problems thoroughly before looking at these solutions. Simply reading the solutions without thinking about the problems will rob you of the practice required to be able to solve complicated problems on your own. You will perform poorly on the exam if you simply attempt to memorise solutions to these problems. Thinking about a problem, even if you do not solve it will greatly increase your understanding of the underlying concepts.

Problems

Problem 1. Use Dijkstra's algorithm to determine the shortest paths from vertex s to all other vertices in this graph. You should clearly indicate the order in which the vertices are visited by the algorithm, the resulting distances, and the shortest path tree produced.

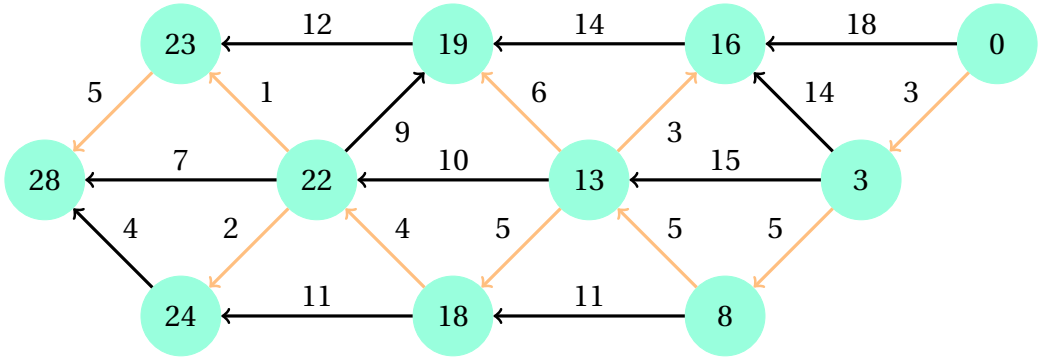


Solution

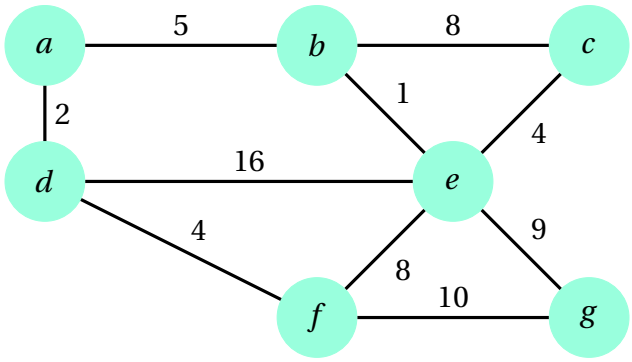
Vertices are visited in the order:

$$s, y, x, v, w, u, t, z, q, r, p$$

The shortest path tree, with vertices labelled by their distances is shown

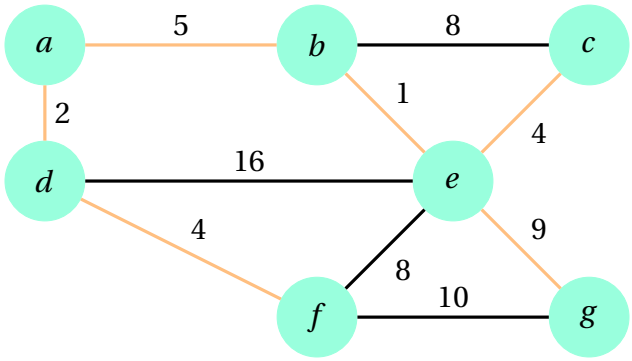


Problem 2. Show the steps taken by Prim's and Kruskal's algorithms for computing a minimum spanning tree of the following graph. Use vertex a as the root vertex for Prim's algorithm. Make sure that you indicate the order in which edges are selected, not just the final answer.



Solution

For Prim's algorithm, we begin at vertex a and select edges in the following order: (a,d), (d,f), (a,b), (b,e), (e,c), (e,g). We end up with the following minimum spanning tree.



Kruskal's algorithm will select edges in the following order: (b,e), (a,d), (d,f), (e,c), (a,b), (e,g). Note that (d,f) and (e,c) could be selected in either order since they have the same weight. The final minimum spanning tree is the same one as obtained by Prim's.

Problem 3. Consider the following state of a union-find data structure, using the union by size heuristic, as described in lectures.

ID	0	1	2	3	4	5	6	7	8
Parent	-1	2	-3	4	7	-1	7	-4	2

Determine the state of the array after each the following operations are executed (in order). If two trees are the same size, assume the first argument to `union()` will be chosen as the root:

1. `union(3, 6)`
2. `union(0, 5)`
3. `union(5, 3)`

Solution

1. No change
2. `Parent[0] = -2, Parent[5] = 0`
3. `Parent[0] = 7, Parent[7] = -6`