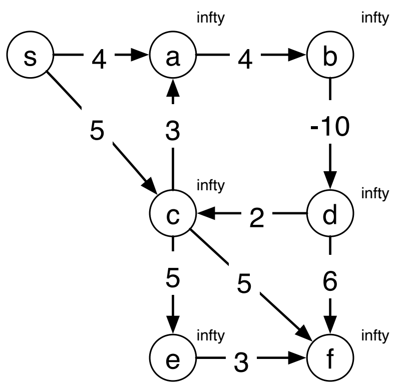
Homework 10 Solutions

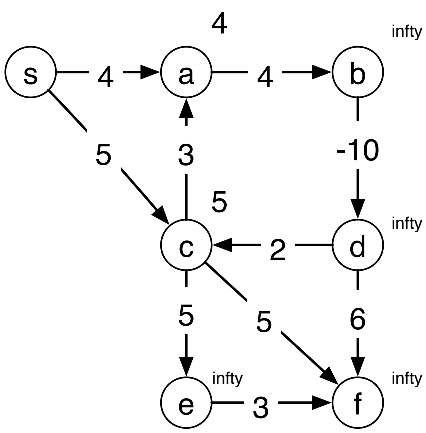
Problem 1:



Priority Queue:

[(a, ∞), (b, ∞), (c, ∞), (d, ∞), (e, ∞), (f, ∞)]

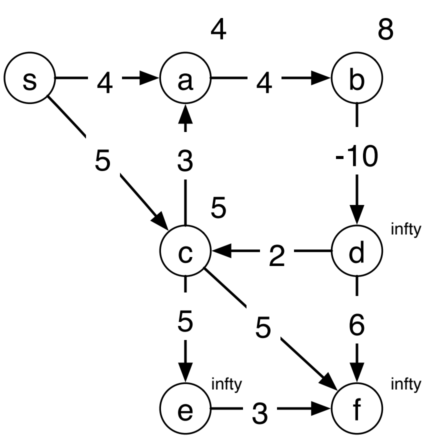
Processing *s:*



Priority Queue:

[(a, 4), (c, 5), (b, ∞), (d, ∞), (e, ∞), (f, ∞)]

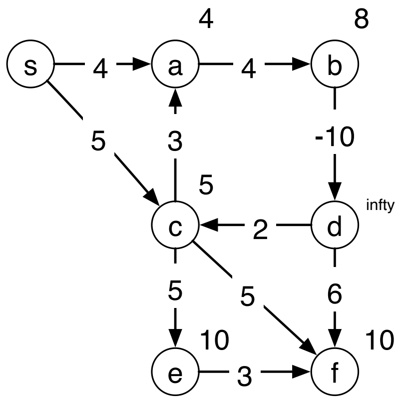
Processing *a*:



Priority Queue:

[(c, 5), (b, 8), (d, ∞), (e, ∞), (f, ∞)]

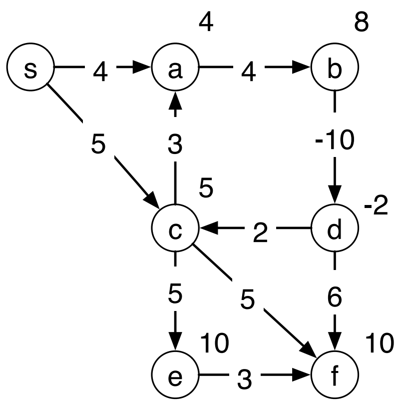
Processing *c*:



Priority Queue:

[(b, 8) (e, 10), (f, 10), (d, ∞)]

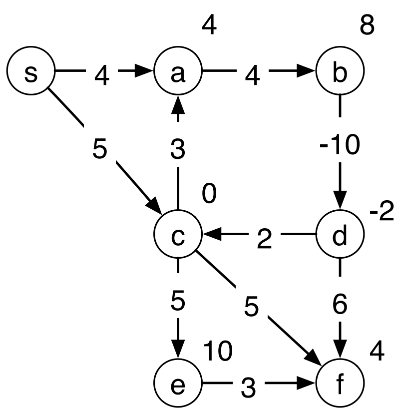
Processing *b*:



Priority Queue:

[(d,-2), (e, 10), (f, 10)]

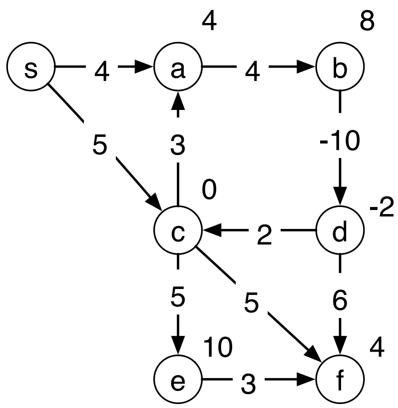
Processing *d*:



Priority Queue:

[(f,4), (e, 10)]]

Processing *f*:
no distance changes, because there are no edges leaving *f*



Priority Queue:

[(e, 10)]]

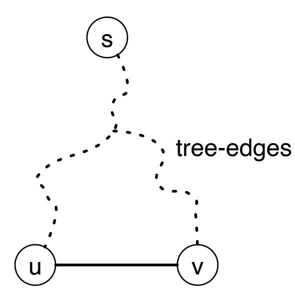
The algorithm now stops because we have reached the last vertex.

As this example shows, Dijskstra will not discover negative edge cycles.

Problem 2:

The Euclidean distance is the shortest path length between two points. Any route from any node to the goal node is a route in the plane.

Problem 3:



(a) If (u, v) is a back-edge, then at the time we consider it, u is in the DFS-tree and v is in the DFS-tree. Since trees are connected, there is another path in the graph from u to v. Therefore, (u, v) cannot be a connector. See diagram on the right.

(b) If a tree edge (u, v) is a connector, then removing it from the graph results in two different components. Therefore, there cannot be any edge from the descendants of v to an ancestor of u.

Conversely, if there is no edge from a descendant of v to an ancestor of u, then removing (u, v) breaks the graph into two different connected components.